

**SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE**

DIPLOMSKI RAD

**PROJEKTIRANJE AUTONOMNOG
ROBOTA KATEGORIJE RESCUE MAZE**

IVAN SIČAJA

Split, rujan 2021.



SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE



Diplomski studij: **Automatika i sustavi**
Smjer/Usmjerenje: /
Oznaka programa: 210
Akademska godina: 2020./2021.

Ime i prezime: **Ivan Sičaja**
Broj indeksa: 659-2019

ZADATAK DIPLOMSKOG RADA

Naslov: **Projektiranje autonomnog robota kategorije Rescue Maze**

Zadatak: Upoznati se s postupcima projektiranja robota prema kriterijima definiranim propisima Robo Cup Junior Rescue Maze 2020. Opisati kriterije za odabir svih komponenti, kao i načela funkcionalnosti pojedinih elektroničkih komponenti, programske logike, korištenih programskih jezika i okruženja, te cijelog procesa proizvodnje mehanike tijela robota. Eksperimentom verificirati postignute funkcionalnosti robota, izvesti odgovarajuće zaključke i opisati moguća poboljšanja u budućem radu.

Rad predan:

Predsjednik
Odbora za diplomski rad:

Mentor:

prof. dr. sc. Mojmil Cević

prof. dr. sc. Mirjana Bonković

IZJAVA

Ovom izjavom potvrđujem da sam diplomski rad s naslovom PROJEKTIRANJE AUTONOMNOG ROBOTA KATEGORIJE RESCUE MAZE pod mentorstvom prof. dr. sc. MIRJANE BONKOVIĆ pisao samostalno, primijenivši znanja i vještine stečene tijekom studiranja na Fakultetu elektrotehnike, strojarstva i brodogradnje, kao i metodologiju znanstveno-istraživačkog rada, te uz korištenje literature koja je navedena u radu. Spoznaje, stavove, zaključke, teorije i zakonitosti drugih autora koje sam izravno ili parafrazirajući naveo u diplomskom radu citirao/la sam i povezao s korištenim bibliografskim jedinicama.

Student:

Ivan Sičaja

SADRŽAJ

1	UVOD	1
2	RESCUE MAZE -ANALIZA PRAVILA I OKRUŽENJA.....	2
3	ODABIR SOFTVERSKJE I HARDVERSKJE STRUKTURE ROBOTA.....	5
3.1	Odabir softverske strukture robota	5
3.2	Odabir hardverske strukture robota	6
4	CJELOKUPAN POSTUPAK IZRADE ROBOTA	8
4.1	Teensy 3.5.....	8
4.1.1	Načini programiranja mikrokontrolera Teensy 3.5	14
4.2	Raspberry Pi 4 model B.....	16
4.2.1	Prvo pokretanje računala Raspberry Pi model 4 B	19
4.2.2	Komandni terminal i instalacija potrebnih modula	21
4.3	Hardverske komponente robota.....	22
4.3.1	Baterija	22
4.3.2	Električni motori	24
4.3.3	Kontroler motora	25
4.3.4	Senzori udaljenosti	28
4.3.5	PDB (eng. Power Distribution Board)	29
4.3.6	Razvod žica	31
4.3.7	Žiroskop	32
4.3.8	Senzor boje.....	34
4.3.9	Termalna kamera.....	37
4.3.10	Raspberry Pi kamera	40
4.3.11	Multi-kamera adapter	41
4.3.12	Zvučnik.....	43
4.3.13	Ledica	44
4.3.14	Solenoid.....	45
4.3.15	Paket pomoći	47
4.3.16	Adapter za Teensy 3.5	47
4.3.17	Raspored komponenti.....	48
4.4	Izrada tijela robota	49
4.4.1	Dizajniranje i priprema 3D modela za postupak 3D printanja.....	49
4.4.2	Postupak 3D printanja	54

4.4.3	Komponente robota koje čine tijelo robota	54
4.5	Električna shema robota	60
4.6	Cjelokupna programska logika robota.....	61
4.6.1	Obrada slike.....	62
4.7	Mehanička rješenja	70
4.7.1	Izrada kotača	70
4.7.2	Izrada amortizirajuće osovine	72
4.7.3	Sustav za dostavljanje paketa pomoći.....	75
5	EKSPERIMENTALNA ISTRAŽIVANJA	77
6	ZAKLJUČAK.....	80
7	POPIS LITERATURE	84
	SAŽETAK.....	87
	KLJUČNE RIJEČI	87
	SUMMARY	87
	KEY WORDS	87
	DODATAK A -PROGRAMSKI KOD	87

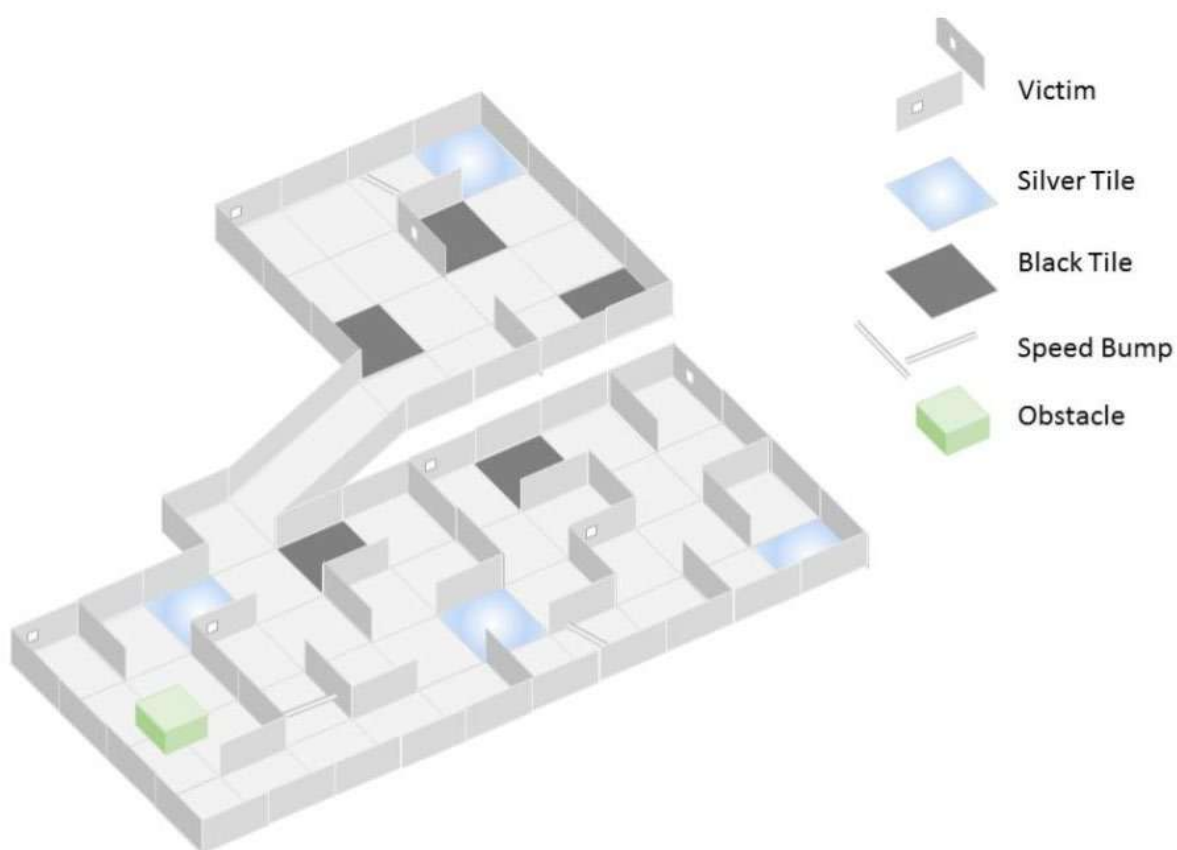
1 UVOD

Zadatak ovog diplomskog rada je projektirati mobilnog robota kategorije Rescue Maze, definirane pravilima Robo Cup Junior natjecanja. Projekt je uključivao osmišljavanje i izradu mehanizama za realizaciju potrebnih funkcionalnosti kao i samog tijela robota, definiranje potrebnog skupa elektroničkih komponenti, te implementaciju programskog rješenja kojim se robot kvalificira za natjecanje u spomenutoj kategoriji. To znači da robot autonomno manevrira u svojem okruženju, sposoban je prepoznati određene objekte - „žrtve“, odnosno mete, i dostaviti im potrebnu pomoć. Okruženje robota je labirint, a detalji kriterija koje robot treba ispunjavati propisani su *Robo Cup Junior Rescue Maze 2020* pravilnikom čija je poveznica dostupna u referencama rada [1].

Kako bi praćenje napisanog sadržaja bilo što razumljivije, rad je podijeljen na šest poglavlja. Nakon uvodnog dijela, dan je opis kriterija natjecanja i karakteristika okruženja prema kojima se izvodi projektiranje robota. Zatim je posebna pažnja posvećena načinu odabira odgovarajuće softverske i hardverske strukture te je, u četvrtom poglavlju, detaljno opisan cjelokupan postupak izrade robota. Funkcionalnosti i rezultati eksperimentalnih istraživanja opisani su u petom poglavlju, a karakteristična opažanja i zaključci detaljno su opisani u posljednjem, šestom poglavlju.

2 RESCUE MAZE -ANALIZA PRAVILA I OKRUŽENJA

Cijeli labirint izrađen je od polja dimenzija 30 cm x 30 cm. Na horizontalnim poljima mogu se nalaziti prepreke visine maksimalno 2 cm, te krhotine visine do najviše 1 cm. Polja mogu biti obojana jednom od tri različite boje: crnom, reflektivno sivom ili bijelom bojom. Kroz crna polja robot smije ići ali ih ne smije prijeći. Dozvoljeno je kročiti na crna polja kako bi robot detektirao boju polja, te nakon što je robot ustanovio da je boja polja crna, ne smije se rotirati dok se nalazi na crnom polju, odnosno jedina dozvoljena radnja mu je pravocrtni povratak na prethodno polje. Reflektivna siva polja služe kao točke napretka (eng. checkpoint), odnosno polja s kojih se robot može ponovno pokrenuti ako se tijekom obavljanja zadatka dogodi nekakav mehanički ili softverskog kvar, kako ne bi cijeli zadatak morao obavljati ispočetka. Robot se može ponovno pokrenuti samo sa reflektivno sivih polja koja je uspješno prepoznao. Uz navedene vrste polja i prepreka koje se mogu nalaziti na njima, postoje i polja pod nagibom od maksimalno 25°, koja vode robot na dio labirinta koji se nalazi na drugoj visinskoj razini.



Slika 2.1 Model Labirinta [1]

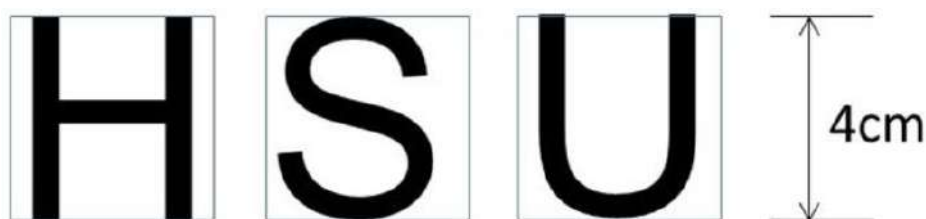
Nakon nabrojanih vrsta horizontalnih polja labirinta i prepreka koje se na njima mogu nalaziti, vrijeme je upoznati se i s objektima koje robot treba autonomno prepoznati.

U labirintu se mogu nalaziti tri vrste objekata koje robot treba moći identificirati, a to su:

- objekti u obliku slova
- objekti u obliku boja
- temperaturni objekti.

Objekti u obliku slova su visine 4 cm, predstavljeni su slovima s fontom nalik na „Arial“ te imaju sljedeće značenje:

- a) slovo „H“ (eng. Harmed): predstavlja „ugroženu“ žrtvu te ju je potrebno identificirati i dostaviti joj dva paketa pomoći
- b) slovo „S“ (eng. Stabile): predstavlja „stabilnu“ žrtvu te ju je potrebno identificirati i dostaviti joj jedan paket pomoći
- c) slovo „U“ (eng. Unharmed): predstavlja „neozlijeđenu“ žrtvu te ju je potrebno samo identificirati



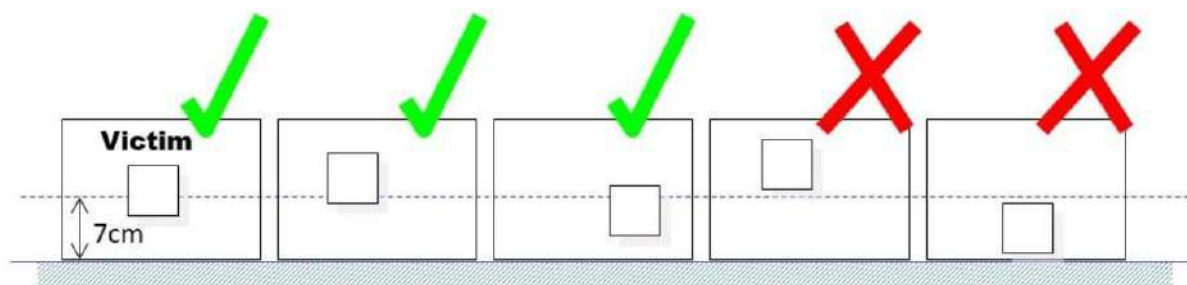
Slika 2.2 Prikaz objekata u obliku slova [1]

Objekti u obliku boja predstavljeni su trima bojama površine 4 cm x 4 cm:

- a) Crvena boja: predstavlja „ugroženu“ žrtvu te ju je potrebno identificirati i dostaviti joj dva paketa pomoći
- b) Žuta boja: predstavlja „stabilnu“ žrtvu te ju je potrebno identificirati i dostaviti joj jedan paket pomoći
- c) Zelena boja: predstavlja „neozlijeđenu“ žrtvu te ju je potrebno samo identificirati

Temperaturna „žrtva“ simulira temperaturu čovjeka, a predstavlja ju izvor topline površine 16 cm² kojem je temperatura u rasponu od 28°C - 40°C. Temperaturnu žrtvu potrebno je identificirati te joj dostaviti jedan paket pomoći.

Sve navedene „žrtve“ se nalaze na vertikalnim pločama labirinta, tj. zidovima čija je minimalna visina 15 cm, a svaka „žrtva“ ima će položaj takav da dodiruje pravac na zidu labirinta, koji se nalazi 7 cm iznad površine na kojoj se nalazi robot i paralelan s površinom na kojoj se nalazi robot.



Slika 2.3 Prikaz mogućih položaja objekata na zidovima labirinta[1]

Robot ne smije imati niti jedan senzor „koji vidi“ iznad 15 cm, gledajući od horizontalne ploče na kojoj se robot nalazi. Ukupna visina robota ne smije prelaziti 30 cm, dok je minimalna širina kroz koju robot treba proći 20 cm.

Minimalni volumen paketa pomoći je 1 cm^3 , a maksimalan broj paketa pomoći koje robot može nositi je dvanaest.

Za komunikaciju je zabranjeno korištenje Wi-Fi protokola, dozvoljen je Bluetooth druge i treće klase. Također je dozvoljeno korištenje lasera prve i druge klase.

Uz sve navedeno cjelokupan dizajn robota, hardverska i softverska rješenja moraju biti potpuno autentična.

Važno je naglasiti da u navedenom pravilniku postoje kriteriji čija realizacija nije predmet ovog diplomskog rada.

3 ODABIR SOFTVERSKJE I HARDVERSKJE STRUKTURE ROBOTA

Jedan od najbitnijih i najzahtjevnijih koraka u izradi ovog projekta je odabir odgovarajuće hardverske i softverske arhitekture na kojoj će robot biti realiziran. Odabir arhitektura ne smije biti pogrešan jer će uzrokovati relativno velike financijske i vremenske gubitke, stoga je potrebno izvršiti jako puno istraživanja, kako bi sve komponente i programski protokoli bili međusobno kompatibilni, i bili u mogućnosti ispuniti postavljene zahtjeve. Ovaj dio zahtjeva jako puno uloženog vremena jer je već u ovom dijelu potrebno znati kako će funkcionirati svaki dio robota u datom trenutku, te nam ostaje još „samo“ isprogramirati i električki spojiti sve komponente i projekt je završen. Naravno svi detalji se nikada ne mogu predvidjeti, **stoga je iznimno važno maksimalno se posvetiti ovoj fazi izrade projekta kako bi naknadne poteškoće koje se budu pojavljivale u realizaciji projekta bile lakše riješive.**

3.1 Odabir softverske strukture robota

Da bi znali koju strukturu odabrati prvo moramo znati koje funkcionalnosti robot treba posjedovati.

Kao osnovne funkcionalnosti koje robot treba posjedovati su: kretanje robota, očitavanje i obrada podataka koji dolaze s raznih senzora, programska izrada mehanizma za dostavljanje paketa pomoći itd. Platforma koja omogućuje sve navedeno je Arduino platforma koja uz programski jezik Arduino, nudi potpuno besplatno programsko okruženje Arduino IDE te mnoštvo raznih senzora, modula, raznih dodataka i protokola koji omogućavaju realizaciju gotovo svega što možete zamisliti. Arduino programski jezik predstavlja blagu modifikaciju C++ programskog jezika, koji je jedan od trenutno najpopularnijih programskih jezika na svijetu, te je shodno tomu C++ programski jezik koji se uči u velikom broju srednjih škola i fakulteta. Budući da je Arduino programska platforma izuzetno popularna u području edukativne robotike i općenito mehatronike, na internetu je vrlo jednostavno pronaći mnoštvo literature, raznih projekata s cjelokupnim hardverskim i softverskim objašnjenjima, raznih biblioteka koje uvelike skraćuju vrijeme pisanja koda, stoga Arduino programska platforma predstavlja gotovo idealan izbor za jednu od platformi koju ćemo koristiti na robotu.

Uz elektroničke funkcionalnosti koje će na robotu obavljati Arduino programska platforma, tu su i nešto sofisticiraniji zadatci koje robot treba obavljati kao što su prepoznavanje slova i boja u svom okruženju. Prepoznavanje slova jako je složen postupak, za koji se često koriste različiti postupci umjetne inteligencije odnosno strojnog učenja, konkretnije korištenje konvolucijskih neuralnih mreža čija je implementacija podržana u programskom jeziku Python. Uz podržavanje strojnog učenja, Python je prema velikom broju izvora trenutno najpopularniji programski jezik na globalnoj razini te omogućuje korištenje mnoštva modula za obradu slike pomoću kojih se na relativno jednostavan način izvornu sliku može obraditi u željeni oblik. Popularnost nekog programskog jezika za sobom povlači i mnoštvo literature što je bio ponovno presudan kriterij za odabir programskog jezika kojeg ćemo koristiti za optičko prepoznavanje objekata. Uz sve navedeno Python je programski jezik koji je vrlo optimiziran, te je sintaktički vrlo jednostavan za korištenje.

3.2 Odabir hardverske strukture robota

Nakon odabiranja softverske arhitekture potrebno je odabrati i hardversku arhitekturu koja na prihvatljiv način podržava odabrani softver.

Ponovno krenimo s Arduino platformom pomoću koje realiziramo mehatroničke zadatke. Ukratko, robot se treba kretati kroz labirint, očitavati i obrađivati podatke prikupljene s različitih senzora, te konstantno donositi niz pravilnih zaključaka na osnovu informacija koje dobiva sa senzora kao i realizaciju raznih regulatora koji na određeni način obrađuju informacije sa senzora (npr. regulator brzine robota ovisno o nagibu) što je većom mogućom brzinom. Zbog prilično velikog broja elektroničkih komponenti koji se koriste, kao i želje da se podatci očitavaju i obrađuju dovoljno brzo, najbitniji kriteriji za odabir odgovarajućeg mikrokontrolera je broj izvoda s popratnim funkcionalnostima, adekvatna procesorska snaga te popularnost mikrokontrolera koja sa sobom donosi mnoštvo literature što je svakako jedan od najbitnijih kriterija. Zbog popularnosti Arduino platforme i dostupnih „open-source“ rješenja, zanimljive su snažnije izvedbe mikrokontrolera koje podržavaju Arduino platformu, a proizvode ih kompanije kao što su Teensy, Espressif, SparkFun i sl.

Tablica 3.1 Usporedba karakteristika određenih mikrokontrolera [2]

SPECIFIKACIJE/MIKROKONTROLER	ESP32	ESP8266	ARDUINO UNO	TEENSY 3.5
Broj jezgri	2	1	1	1
Arhitektura	32 Bit	32 Bit	8 Bit	32 Bit
CPU Frekvencija	160 MHz	80 MHz	16 MHz	120 MHz
WiFi	DA	DA	NE	NE
BLUETOOTH	DA	NE	NE	NE
RAM	512 KB	160 KB	2 KB	256 KB
FLASH Memorija	16 MB	16 MB	32 KB	512 KB
GPIO Pinovi	36	17	14	64
Protokoli	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART	SPI, I2C, UART, I2S, CAN
ADC Pinovi	18	1	6	27
DAC Pinovi	2	0	0	2

Iako Teensy 3.5 mikrokontroler u odnosu na ESP-32 na određenim poljima ima slabije rezultate, najzanimljiviji je podatak o broju digitalnih i analognih pinova određenog mikrokontrolera koji može prihvatiti očekivano veliki broj senzora i modula koje Rescue Maze robot koristi i koji nude dodatni prostor za daljnje hardverske nadogradnje i optimizacije. Frekvencija i broj jezgri procesora svakako je bitan, što ovdje ne igra veliku ulogu jer je frekvencija taktnog signala od 120 MHz i više nego zadovoljavajuća, kao i 512 KB „Flash“, 256KB radne memorije, čiji je detaljniji opis funkcionalnosti dostupan na referenci [2]. Mogućnost korištenja Wi-Fi i Bluetooth protokola također za ovaj projekt nije bitna jer se ovi protokoli ne koriste u komunikaciji podsustava robota. Bitnije je da se može istovremeno

koristiti dovoljan broj digitalnih i analognih pinova, što i nije slučaj za ESP-32 koji u korištenju određenih režima rada, određene GPIO pinove stavlja izvan funkcije. Drugi bitan kriterij je taj da mikrokontroler podržava SPI, I2C i UART protokole koji su standard za uspostavljanje komunikacije s drugim mikrokontrolerima, senzorima ili računalima. Teensy 3.5 podržava navedene protokole, ima značajno više GPIO pinova od navedenih mikrokontrolera, a k tomu su mu dimenzije vrlo kompaktne u odnosu na npr. Arduino Mega mikrokontroler kojeg također krasi veliki broj pinova, ali su mu mana vrlo glomazne dimenzije koje je jako teško uklopiti u projektne ciljeve. Zbog svih navedenih razloga, odabran je Teensy 3.5 kao mikrokontroler koji će upravljati većinom mehatroničkih radnji na robotu kao što su kretanje robota, očitavanje digitalnih i analognih senzora, razne obrada podataka i sl. Uz Teensy 3.5 postoje i dosta naprednije verzije Teensy mikrokontrolera kao što su Teensy 3.6 i Teensy 4.1, koji dolaze s identičnim dimenzijama ali s poboljšanim funkcionalnostima tako da se buduće nadogradnje mikrokontrolera, ako budu potrebne, mogu realizirati s mikrokontrolerom iz iste obitelji, što osigurava visok stupanj kompatibilnosti sa svim postojećim hardverskim i softverskim rješenjima. Detaljnija objašnjenja funkcionalnosti i načina programiranja mogu se pronaći u poglavlju „4.1 Teensy 3.5“.

Nakon odabira mikrokontrolera koji će obavljati većinu mehatroničkih radnji na robotu potrebno je obarati hardver koji će obrađivati podatke vezane za obradu slike. Kao što je objašnjeno u sljedećem podpoglavlju. Programski jezik koji se koristiti za obradu slike je Python, stoga je važno da odabrani hardver podržava Python programski jezik. Uz Python, bitno je da na hardver bude omogućeno povezivanje barem dvije kamere istovremeno, kao i da posjeduje dovoljnu procesorsku snagu. Ponovo najbitniji kriterij za odabir hardvera je postojanje mnoštva literature na koju se možemo referirati. Sve navedene kriterije ispunja računalno Raspberry Pi.

Uz sve navedeno vrlo je kompaktnih dimenzija što je vrlo bitan kriterij pri odabiru gotovo svih komponenti robota, podržava UART, SPI i I2C komunikacijske protokole što omogućava komunikaciju s Teensy 3.5 mikrokontrolerom, te je relativno mali potrošač električne energije što je također vrlo bitan faktor, ako se pretpostavi da će robot imati vlastiti spremnik energije. Cjelokupan postupak instalacije operativnih sustava, instalacije potrebnih aplikacija i Python modula, realiziranje načina interakcije s Raspberry Pi računalom biti će detaljnije objašnjena u nastavku, točnije u podpoglavlju „4.2 Raspberry Pi 4 Model B“.

4 CJELOKUPAN POSTUPAK IZRADE ROBOTA

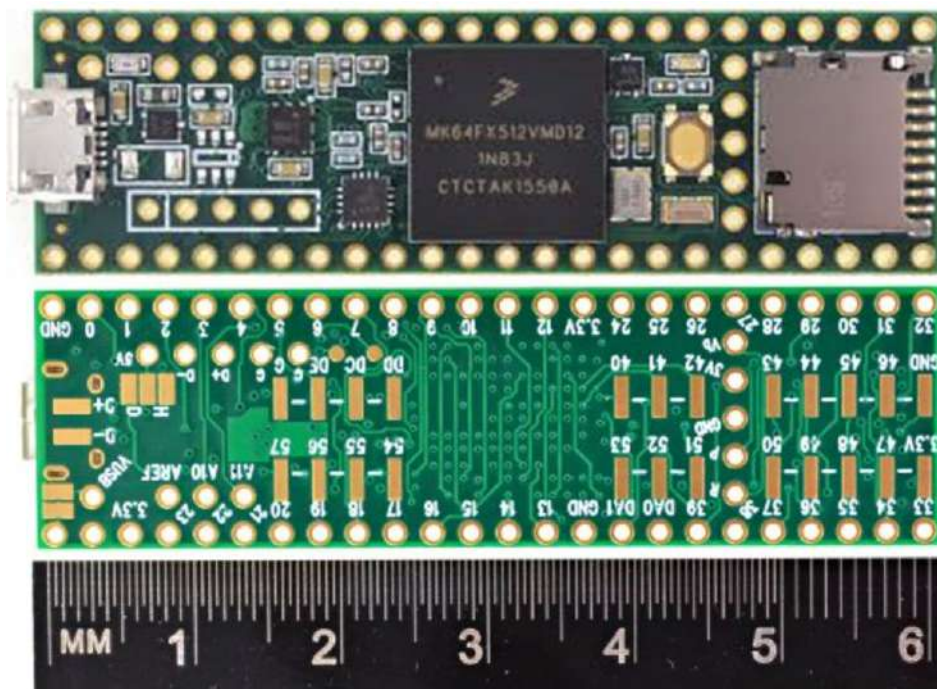
U ovom poglavlju detaljno su objašnjeni postupci kojima smo se koristili kako bi ispunili zadane kriterije te problematiku s kojom smo se tijekom cjelokupnog postupka susreli.

Budući da je postupak vrlo složen te dolazi do čestog ispreplitanja između odabranih softverskih i hardverskih platformi te mehaničkog i nemehaničkog dijela koji svaki zasebno povlače svoju problematiku, cjelokupan postupak je odvojen na dijelove koji se mogu (čak i fizički) izdvojiti kao zasebne cjeline robota, i to: funkcionalnosti za koje je zadužen Teensy, funkcionalnosti za koje je zadužen Raspberry Pi, zahtjeve i problematiku mehaničkih dijelova, kao i cjelokupan postupak izrade tijela robota korištenjem 3D printera te električno spajanje svih navedenih komponenti robota u funkcionalnu cjelinu.

4.1 Teensy 3.5

U nastavku ovog poglavlja predstavljena je građa i mogućnosti mikrokontrolera Teensy 3.5 čija je funkcija vezana za koordinaciju mehatroničkih radnji na robotu kao što su kretanje, očitavanje senzora i sl.

Prikaz ovog mikrokontrolera dan je na sljedećoj slici.



Slika 4.1 Prikaz gornje i donje strane mikrokontrolera Teensy 3.5 [3]

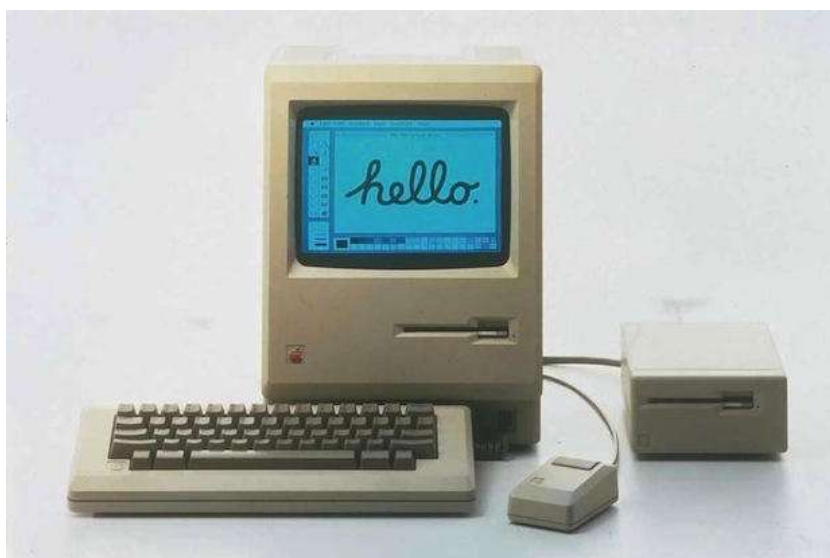
Osnovne karakteristike su dane u sljedećoj tablici.

Tablica 4.1 Prikaz karakteristika mikrokontrolera Teensy 3.5

PROCESOR	ARM Cortex-M4 na 120 MHz
PROGRAMSKA MEMORIJA	512KB
RADNA MEMORIJA	256KB
EEPROM MEMORIJA	4K
BRZINA SERIJSKE KOMUNIKACIJE	12 Mbit/sec
BROJ DIGITALNIH PINOVA	64
BROJ PWM IZLAZA	20
BROJ ANALOGNIH ULAZA	27
BROJ ANALOGNIH IZLAZA	2
BROJ KANALA ZA SERIJSKU KOMUNIKACIJU	6
BROJ KANALA ZA SPI KOMUNIKACIJU	3
BROJ KANALA ZA I2C KOMUNIKACIJU	2
BROJ KANALA ZA CAN BUS KOMUNIKACIJU	2
BROJ DIGITALNIH AUDIO KANALA	1
KANALA ZA KOMUNIKACIJU SA SD KARTICOM	1
BROJ DMA KANALA	16
BROJ RTC KANALA	1

Navede informacije o mikrokontroleru ništa ne znače ukoliko im ne znamo svrhu te ih ne usporedimo s nečim prepoznatljivim kako bismo dobili osjećaj što dati podatak zapravo predstavlja.

Teensy 3.5 je mikrokontroler građen od ARM Cortex-M4 procesora, frekvencije taktnog signala od 120 MHz. Jednostavnije rečeno za vremenski period od jedne sekunde ovaj mikroprocesor može izvršiti 120 miliona radnji. Usporedi li se to s procesorom nekog modernog današnjeg laptopa, kao što je na primjer laptop Lenovo Ideapad L340-15IRH 81LK0060SC, čiji procesor ima 6 jezgri i frekvenciju taktnog signala prosječno 3 GHz, procesor M4 ima samo jednu jezgru koja radi otprilike sporije 25 puta od navedenog laptopa. Cijena navedenog laptopa je 7000 HRK, dok je cijena mikrokontrolera Teensy 3.5 oko 150 HRK. Iako se navedeni podatci ne mogu uspoređivati jer se radi o dva uređaja potpuno različite namjene, vrijedno ih je spomenuti kako bi se podatci iz prethodne tablice bili intuitivnije razumljiviji. Sljedeći podatak je programska memorija u kojoj je pohranjen programski kod koji definira način rada mikrokontrolera. Ukoliko sam programski kod zauzima više od 512 KB memorije, prijenos koda na mikrokontroler neće biti moguć. U usporedbi s današnjim računalima to je otprilike 1 000 000 puta manje memorije za pohranu od raspoloživog memorijskog prostora navedenog laptopa, čija količina memorije za pohranu iznosi 512 GB, odnosno M4 ima identičnu količinu memorije za pohranu kao i Macintosh 512K, računalo koje se proizvodilo od 1984 do 1986. godine, te imalo kako mu i sam naziv govori 512 KB memorije za pohranu.



Slika 4.2 Prikaz izgleda računala Macintosh 512K [4]

Radna memorija (eng. Random-Access Memory) predstavlja količinu memorije koju mikrokontroler koristi u radu, odnosno za definiranje varijabli, pozive funkcija, razna alociranje i dealociranje memorije, te u odnosu na navedeno računalo koje posjeduje 8 GB radne memorije, 256 KB je 31 250 puta manja količina. EEPROM memorija je memorija koja služi da pohranu određenih podataka koji ostaju spremljeni i onda kada je mikrokontroler ostane bez električne energije, odnosno možemo ga smatrati kao nekom vrstom memorijske kartice za mikrokontrolere. EEPROM memorija ovog mikrokontrolera iznosi 4KB. Postavlja se pitanje kolika je to zapravo količina ove i svih navedenih vrsta memorije do sada. Prema načinu kodiranja znakova s tipkovnice suvremena računala, ukoliko se koristi EASCII standard (eng. Extended American Standard Code for Information Interchange) za upisivanje jednog znaka potrebno je alocirati količinu memorije od osam bita [b] što je jednako jednom bajtu [B]. Ako svaki znak zauzima 1B, u 4 KB može se pohraniti 4000 znakova. Međutim, ovisno o tipu varijable zauzimaju se različite količine memorije. Neki broj tipa int (eng. Integer) neće zauzimati jednaku količinu memorije kao broj zapisan u decimalnom obliku tipa float (eng. Float). Prikaz potrebne količine alokacije memorije za određene tipove podatka u Arduino programskom jeziku prikazan je u sljedećoj tablici.

Tablica 4.2 Prikaz karakteristika određenog tipa varijabli u Arduino Programskom jeziku [2]

Tip podatka	Veličina u Bajtovima	Može sadržavati:
boolean	1	Istina (1) ili laž (0)
char	1	ASCII simbol ili vrijednosti od -128 do 127
unsigned char, byte, uint8_t	1	ASCII simbol ili vrijednosti od 0 do 255
int, short	2	Vrijednosti od -32 768 do 32 767
unsigned int, word, uint16_t	2	Vrijednosti od 0 do 65 535
long	4	Vrijednosti između -2,147,483,648 i 2,147,483,647
unsigned long, uint32_t	4	Vrijednosti između 0 i 4,294,967,295
float, double	4	Decimalne vrijednosti između 3.4028235E+38 i 3.4028235E+38 (NAPOMENA) float i double tipovi podataka na ovoj platformi su istoznačni

Stoga birajući adekvatne tipove varijabli mogu se uštediti značajni postoci memorijskog prostora, posebno ako se radi o jako malim količinama memorije za pohranu kako što je 4 KB, što je količina EEPROM memorije mikrokontrolera Teensy 3.5. Detaljnije postupke funkcioniranja i optimizacije memorije dostupni su u završnom radu [2] Broj digitalnih i analognih pinova naveden je kao ključan čimbenik u odabiru upravo ovog mikrokontrolera u odnosu na mikrokontrolere kao što su ESP-32, Arduino Uno, Teensy 4.1 i sl. Naime ovaj mikrokontroler posjeduje čak 58 digitalnih i 27 analognih pinova, što je značajno više pinova od onih na mikrokontroleru ESP-32, koji je ujedno bio glavni konkurent pored izabrane Teensy platforme, koji posjeduje 48 GPIO pinova. Od tih pinova, 18 je analognih pinova što je značajno manje od 27 analognih pinova koje posjeduje mikrokontroler Teensy 3.5 koji u ovom segmentu dominira i u svojoj platformi. Detaljne karakteristike mikrokontrolera bazirane na Teensy platformi možete vidjeti na sljedećoj tablici.

Tablica 4.3 Usporedba karakteristika različitih Teensy mikrokontrolera [5]

Feature	Teensy 2.0	Teensy++ 2.0	Teensy LC	Teensy 3.2	Teensy 3.5	Teensy 3.6	Teensy 4.0	Teensy 4.1	Units
Price	\$16.00	\$24.00	\$11.65	\$19.80	\$24.25	\$29.25	\$19.95	\$26.85	US Dollars
Processor Core	ATMEGA32U4	AT90USB1286	MKL26Z84VFT4	MK20DX256VLH7	MK64FX512VMD12	MK66FX1M0VMD18	IMXRT1062DVL6	IMXRT1062DVL6	
FPU	AVR	AVR	Cortex-M0+	Cortex-M4	Cortex-M4F	Cortex-M4F	Cortex-M7	Cortex-M7	bits
Rated Speed	16	16	48	72	120	180	600	600	MHz
Overclockable	-	-	-	96	-	240	912	912	MHz
Flash Memory	31.5	127	62	256	512	1024	1984	7936	kbytes
Bandwidth	32	32	96	192	192	411	66	66	Mbytes/sec
Cache	-	-	64	256	256	8192	65536	65536	Bytes
RAM	2.5	8	8	64	256	256	1024	1024	kbytes
EEPROM	1024	4096	128 (emu)	2048	4096	4096	1080 (emu)	4284 (emu)	bytes
Direct Memory Access	-	-	4	16	16	32	32	32	Channels
Digital I/O	25	46	27	34	58	58	40	55	Pins
Breadboard I/O	22	36	24	24	40+2	40+2	24	42	Pins
Voltage Output	5V	5V	3.3V / 5V	3.3V	3.3V	3.3V	3.3V	3.3V	Volts
Current Output	20mA	20mA	5mA / 20mA	10mA	10mA	10mA	10mA	10mA	milliAmps
Voltage Input	5V	5V	3.3V Only	5V Tolerant	5V Tolerant	3.3V Only	3.3V Only	3.3V Only	Volts
Interrupts	4	8	18	34	58	58	40	55	Pins
Analog Input	12	8	13	21	27	25	14	18	Pins
Converters	1	1	1	2	2	2	2	2	
Usable Resolution	10	10	12	13	13	13	10	10	Bits
Prog Gain Amp	1	1	-	2	-	-	-	-	
Touch Sensing	-	-	11	12	-	11	-	-	Pins
Comparators	1	1	1	3	3	4	4	4	
Analog Output	-	-	1	1	2	2	-	-	Pins
DAC Resolution	-	-	12	12	12	12	-	-	Bits
Timers	4 Total	4 Total	7 Total	12 Total	17 Total	19 Total	49 Total	49 Total	
PWM, 32 bit	-	-	-	-	-	-	3	3	
PWM, 16 bit	2	2	3	3	4	6	32	32	
PWM, 8-10 bit	2	2	-	-	-	-	-	-	
Total PWM Outputs	7	9	10	12	20	22	27	31	Pins
PDB Type	-	-	-	1	1	1	-	-	
CMT Type	-	-	-	1	1	1	-	-	
Quadrature Enc	-	-	-	-	-	-	4	4	
LPTMR Type	-	-	1	1	1	1	-	-	
PIT/Interval	-	-	2	4	4	4	4	4	
IEEE1588	-	-	-	-	4	4	4	4	
Systick	-	-	1	1	1	1	1	1	
RTC	-	-	0 **	1 **	1	1	1	1	
Communication									
USB	1	1	1	1	1	2	2	2	
Serial	1	1	3	3	6	6	7	8	
With FIFOs	-	-	-	2	2	2	7	8	
High Res Baud	-	-	-	3	6	5	-	-	
SPI	1	1	2	1	3	3	2	2	
With FIFOs	-	-	1	1	1	1	2	2	
I2C	1	1	2	2	3	4	3	3	
CAN Bus	-	-	-	1	1	2	3	3	
With CAN-FD	-	-	-	-	-	-	1	1	
Digital Audio In	-	-	1	2	2	2	5*	5*	stereo pins
Digital Audio Out	-	-	1	2	2	2	5*	5*	stereo pins
S/PDIF Input	-	-	-	-	-	-	1	1	
S/PDIF Output	-	-	-	0*	0*	0*	1	1	
MQS Output	-	-	-	-	-	-	1	1	
SD Card	-	-	-	-	1	1	1*	1	
Ethernet	-	-	-	-	1*	1*	-	1	uu

Digitalni pinovi najčešće služe za očitavanje digitalnih senzora, uspostavljanje komunikacije između uređaja koji podržavaju UART, SPI i I2C komunikacijske protokole, te za kreiranje PWM signala pomoću kojeg se kontroliraju servo motori ili pak upravljaju istosmjerni motori preko sklopova koje PWM signal može dekodirati. Nešto više o samom PWM signalu dostupno je na referenci [2]. Analogni pinovi uglavnom služe za očitavanje analognih senzora, potencijometara i sl. Veći broj digitalnih i analognih pinova omogućuje spajanje više senzora na naš uređaj, korištenje određenih protokola za komunikaciju s određenim uređajima, te upravljanje većim brojem istosmjernih motora. Uz sve navedeno ovaj mikrokontroler ima

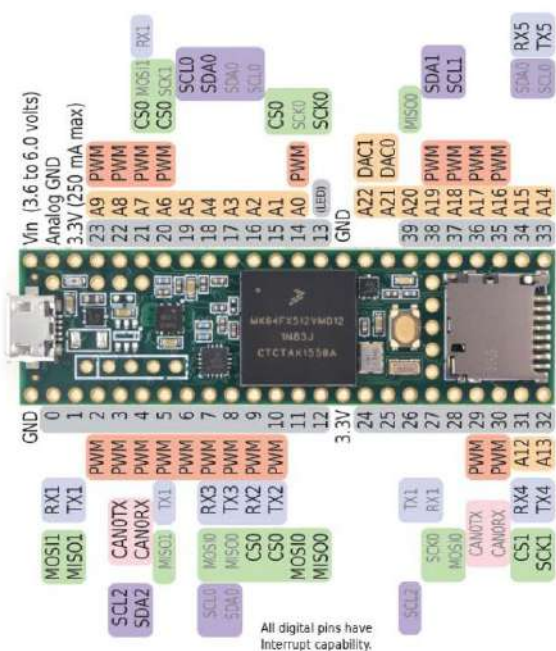
sposobnost korištenja navedenih protokola na više kanala što također u složenim sustavima kao što je robot predstavljen u ovom radu svakako dobro dođe. Osim navedenog, Teensy ima 16 DMA kanala (eng. Direct Memory Access) koji omogućavaju direktnu komunikaciju uređaja sa SRAM memorijom (nije potrebno za koristiti CPU), što omogućava brži odziv sustava. Ukoliko su potrebni podatci o realnom vremenu, nije potrebno ugrađivati dodatni RTC (eng. Real Time Clock) modul već je dovoljno samo na odgovarajuće pinove dovesti napajanje koje će omogućiti da se vrijeme nastavi odbrojavati nakon što prilikom prenošenja koda na mikrokontroler Teensy 3.5 automatski preuzme podatke o trenutnom vremenu s uređaja s kojeg je vršeno prenošenje koda. Također ovaj mikrokontroler ima sposobnost upisivanja i očitavanja podataka s memorijske kartice što svakako može poslužiti, ukoliko EEPROM memorija nije dostatna. Prikaz svih pinova mikrokontrolera Teensy 3.5 sa svojim funkcionalnostima dan je na sljedećoj slici.

Welcome to Teensy® 3.5

32 Bit Arduino-Compatible Microcontroller

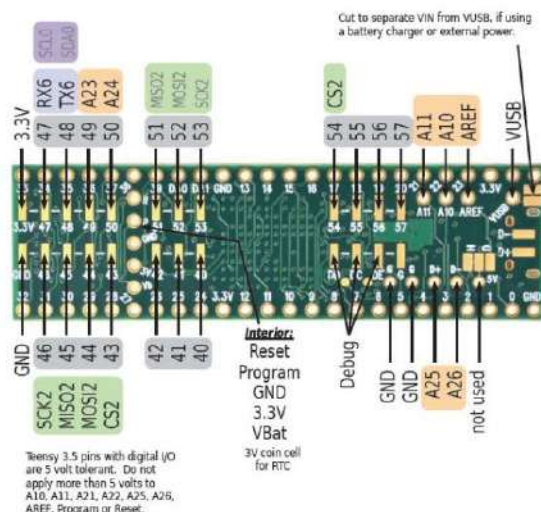
To begin using Teensy, please visit the website & click [Getting Started](#).

www.pjrc.com/teensy



Teensy® 3.5 Back Side

Additional pins and features available on the back side



For solutions to the most common issues and technical support, please visit:

www.pjrc.com/help

Teensy 3.5 System Requirements:
PC computer with Windows 7, 8, 10 or later
or Ubuntu Linux 12.04 or later
or Macintosh OS-X 10.7 or later
USB Micro-B Cable



Slika 4.3 Prikaz oznaka i funkcionalnosti pinova mikrokontrolera Teensy 3.5 [3]

Iz prethodne slike je vidljivo da u originalnom obliku Teensy 3.5 dolazi bez izvoda pinova, odnosno nožica koje omogućuju jednostavno komuniciranje, stoga je pinove potrebno zalemiti na sve pinove koje želimo koristiti. Izgled mikrokontrolera nakon dodavanja „nožica“ prikazan je na sljedećoj slici.



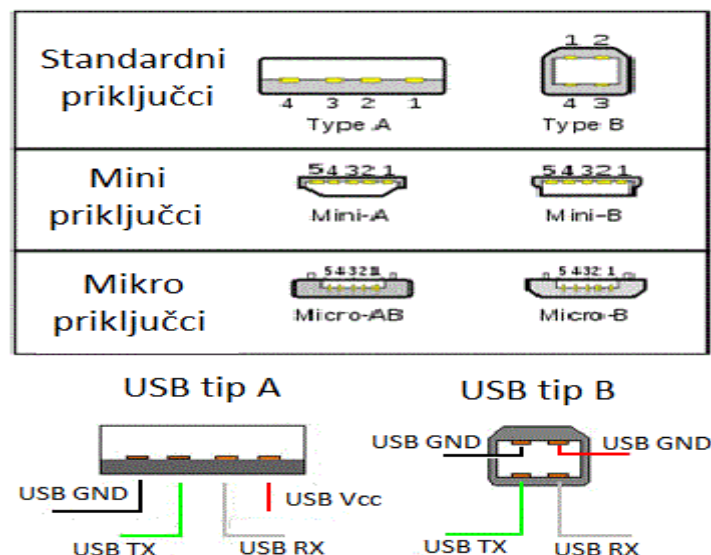
Slika 4.4 Prikaz mikrokontrolera Teensy 3.5 sa zalemljenim izvodima pinova [3]

Nakon pojednostavljenog opisa građe mikrokontrolera, u nastavku je objašnjeno kako programirati navedeni mikrokontroler.

4.1.1 Načini programiranja mikrokontrolera Teensy 3.5

Osnovni način programiranja svih mikrokontrolera koji podržavaju arduino programski jezik je programiranje pomoću službenog i besplatnog programa koji nosi naziv Arduino IDE, koji se može preuzeti sa službenih stranica Arduino-a. Dobre strane ovog softvera su što sadrži vrlo jednostavno sučelje te mnoštvo primjera kodova koji su podijeljeni prema logički smislenim cjelinama, koji uvelike mogu biti od koristi. Veliki nedostatak je nedostatak „IntelliSense-a“ tj, algoritma koji nakon unošenja određene količine znakova sugerira traženu riječ, što najviše dolazi do izražaja kod programa s velikim brojem varijabli i funkcija kada morate biti sigurni da ste korektno unijeli željeni naziv, te nedostatak nekih od standardnih kratica za modificiranje koda, što programiranje u ovom okruženju čini jako sporim.

Ako se želi programirati Teensy pločice u ovom programskom okruženju, potrebno je instalirati program naziva Teensyduino kojeg možete preuzeti sa službene PJRC stranice. Nakon preuzimanja potrebno je slijediti upute koje program ispisiuje ili pak slijediti upute sa službene stranice. Ovaj program zapravo predstavlja „driver“ pomoću kojeg računalo može prepoznati da se na određenom USB portu nalazi upravo Teensy pločica, budući da se Teensy mikrokontroleri programiraju preko serijske veze omogućene USB kabelom.



Slika 4.5 Prikaz često korištenih USB konektora [2]

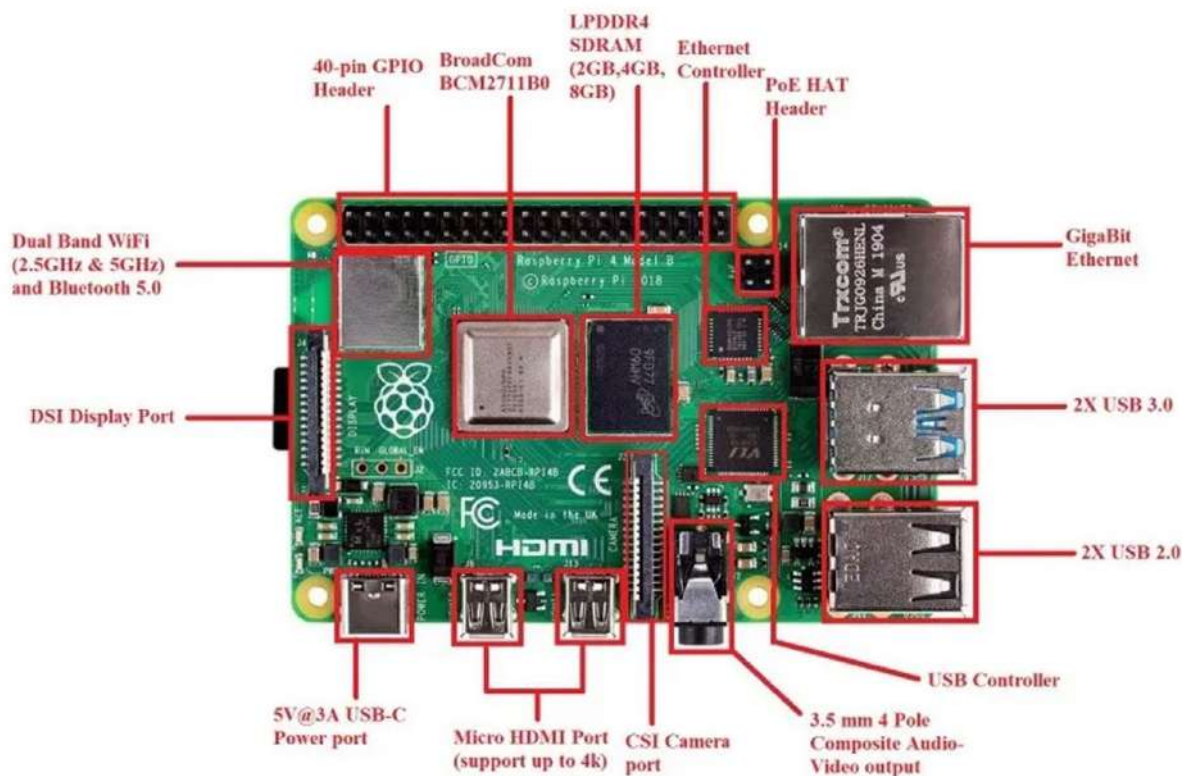
Ukoliko napisani kod nema pogrešaka pri kompajliranju, može se prenijeti na odabrani mikrokontroler. Cjelokupan postupak pisanja jednostavnog Arduino koda, te provjera i prenošenje koda na mikrokontroler više je nego dostupan na platformi YouTube kao i na službenim Arduino i PJRC stranicama.

Druga opcija je korištenje okruženja Platformio IDE koje uz vrlo intuitivno sučelje nudi mogućnost korištenja raznih programerskih kratica za modeliranje koda kao i korištenje Intellisens-a, što uvelike ubrzava pisanje programskog koda. Budući da se za hardverski i softverski složene Arduino projekte gotovo podrazumijeva korištenje većeg broj gotovih biblioteka koje koriste razni moduli, senzori, servo motori ili pak sama mikrokontrolerska pločica, velika većina biblioteka ne dolazi kao standardna u službenom programskom okruženju Arduino IDE, te je svaku potrebno ručno preuzeti i uključiti u projekt, dok Platformio IDE uz sve navedene opcije nudi i opciju za dodavanje određenih biblioteka u projekt koje i nakon preuzimanja te korištenja na drugom računalu ostaju u projektu te nije potrebno svaku biblioteku posebno dodavati ako se radi projekt na nekom drugom uređaju ili pak na projektu radi više osoba, te će im ova opcija biti od velikog značaja. Platformio IDE dolazi kao ekstenzija za programska okruženja kao što su Atom ili Visual Studio Code. Preporuka je svakako korištenje Visual Studio Code okruženja koji se prema dugogodišnjem iskustvu pokazao kao najbolja opcija u ovom području, dok je Platformio IDE ekstenzija na Atom okruženju, prema vlastitom iskustvu, nakon par mjeseci rada na projektu odjednom počela javljati čudne greške prilikom kompajliranja koda, kojima ni nakon nešto manje od dvije godine nije otkriven pravi uzrok, dok je isti kod u Visual Studio Code okruženju uredno kompajliran.

Sada kada su poznate mogućnosti i način programiranja Teensy 3.5 mikrokontrolera vrijeme je upoznati se sa Raspberry Pi platformom.

4.2 Raspberry Pi 4 model B

Raspberry Pi je platforma koja nudi performanse računala u dimenzijama mikrokontrolera što ju čini idealnim izborom za mnoge, kako istraživačke projekte tako i za projekte koji zahtijevaju svakodnevno korištenje hardvera (npr. prijenos videa uživo na Internet stranicu). Za ovaj projekt, najbitnija karakteristika ovog računala je odličan omjer performansi i dimenzija što omogućava realiziranje vrlo složenih softverskih zadataka na vrlo malom volumenu. Za projekt je korišten Raspberry Pi 4 model B čija je građa i mogućnosti ukratko opisana u nastavku. Građa Raspberry Pi model 4 B prikazana je na narednoj slici.



Slika 4.6 Prikaz građe računala Raspberry Pi 4 Model 4 [6]

Uz sliku tu je popis bitnih hardverskih mogućnosti koje su objašnjene u nastavku:

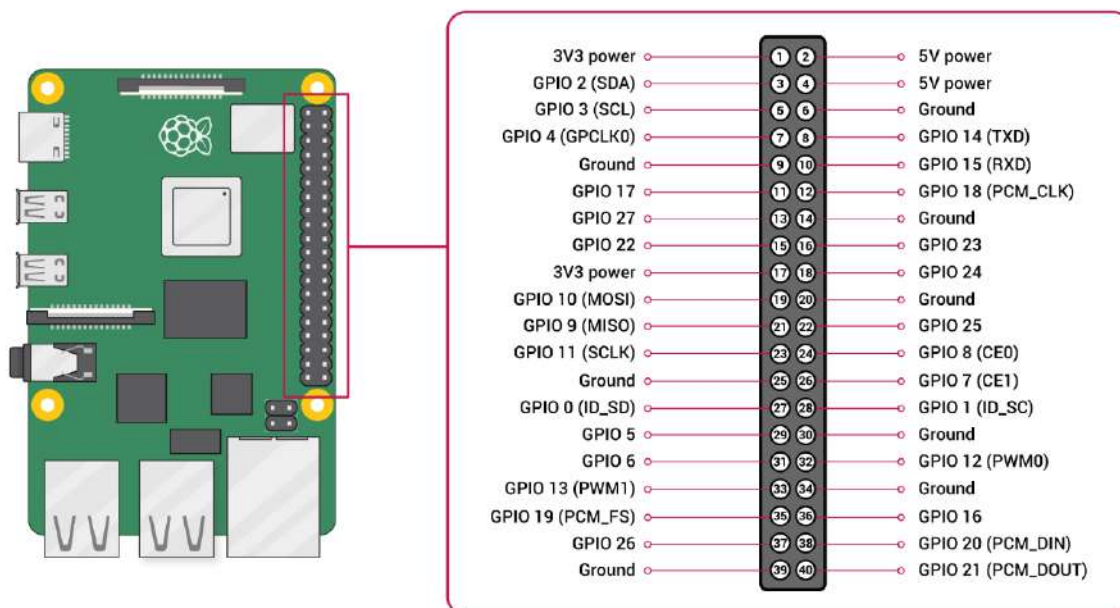
- Broadcom BCM2711, Četvero-jezgri Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB ili 8GB LPDDR4-3200 SDRAM (zavisi o modelu)
- 2.4 GHz i 5.0 GHz IEEE 802.11ac bežični, Bluetooth 5.0, BLE
- Gigabitni Ethernet
- 2 USB 3.0 porta; 2 USB 2.0 porta.
- 2 × micro-HDMI porta (4kp60 podržano)
- 1 x MIPI DSI kanal za ekran

- 1 x MIPI CSI kanal za kameru
- 4-polni stereo audio i kompozitni video port
- Micro-SD port za pohranu podataka i operativnog sustava
- H.265 (4kp60 dekodir), H264 (1080p60 dekodir, 1080p30 enkoder)
- 40 Raspberry Pi GPIO pinova (potpuno kompatibilni sa prethodnim verzijama računala)
- 5V DC - USB-C konektor (minimalno 3A)
- 5V DC - GPIO (minimalno 3A)
- Napajanje preko ethernet kabela – PoE (eng. Power over Ethernet) omogućeno (potreban zaseban PoE adapter)
- Radna temperatura: 0 -50 °C

Kako je navedeno na početku prethodnog popisa, Raspberry Pi 4 Model B posjeduje ARM procesor koji se sastoji od čak četiri jezgre što omogućava stvarni multitasking, odnosno izvršavanje više naredbi paralelno pri čemu bi određena jezgra procesora zadužena za određeni zadatak. Uspoređujući Raspberry Pi Model 4 B s laptopom s kojim je uspoređen mikrokontroler Teensy 3.5, procesor Raspberry Pi računala se sastoji od samo dvije procesorske jezgre manje u odnosu na spomenuto računalo te mu je frekvencija dvostruko manja što predstavlja izvrsne procesorsku moć uzme li se u obzir cijena i dimenzije Raspberry Pi računala. Raspberry Pi Model 4 B, ovisno o verziji dolazi s 2 GB, 4 GB ili čak 8 GB RAM memorije što je uistinu impresivan podatak. Želi li se koristiti Internet, moguće mu je pristupiti preko ugrađenog Wi-Fi modula ili pak preko Gigabitnog Ethernet kabela. Ovo računalo posjeduje čak 4 USB priključka tipa A, od kojih su dva priključka koriste USB 2.0 protokol dok preostala 2 USB priključka koriste USB 3.0 protokol. Navedeni USB priključci imaju identičnu funkciju kao i USB priključci na suvremenim računalima, odnosno mogu se koristiti za priključivanje tipkovnice, miša, USB kamera, očitavanje USB diskova i sl. Moguće je pogledati video u 4K 60 FPS (eng. Frame Per Second) rezoluciji, jer Raspberry Pi Model 4 B posjeduje čak dva micro-HDMI (eng. High Definition Multimedia Interface) izlaza na koje se može povezati ekran te bi se uz korištenje miša i tipkovnice moglo koristiti Raspberry Pi kao računalo opće namjene. Uz HDMI izlaze na Raspberry-ju ne nalazi jedan MIPI DSI izlaz (eng. Mobile Industry Processor Interface Display Serial Interface) kao i MIPI CSI izlaz (eng. Camera Serial Interface) koji služe za povezivanje uređaja (ekrana ili kamere) koji podržavaju ovakve protokole komunikacije. Uz sve navedeno Raspberry Pi računalo posjeduje izlaz za stereo audio i kompozitni video, čitač SD kartice čiji kapacitet ne smije prelaziti 32 GB iz razloga što memorijske kartice s kapacitetom većim od 32 GB se formatiraju pomoću exFAT protokola (eng. File Allocation Table) protokola kojeg bootloader Raspberry Pi Model 4 B računala ne podržava, odnosno podržava FAT16 ili FAT32 protokole kojima se formatiraju kartice memorijskog kapaciteta od 32 GB prema manjim vrijednostima. Memorijska se kartica može koristiti kao glavna ili sporedna memorija računala tj. na memorijsku karticu može se instalirati operativni sustav Raspberry Pi računala. Ono po čemu se ovo računalo razlikuje od „uobičajenih“ računala je 40 pinova opće namjene od koji je 26 programabilno, mogu se koristiti na gotovo identičan način kao i pinovi na Teensy mikrokontroleru, odnosno očitavati vrijednosti senzora, pokretati robota i sl. dok jedina je razlika način na koji se pinovi računala programiraju. Preostali pinovi koji nisu programibilni uglavnom služe kao izvori električne energije ili pak kao kanal za napajanje Raspberry Pi

računala, koje se osim preko USB priključka tipa C, može napajati i preko Ethernet kabela nakon nadogradnje PoE HAT modulom (eng. Power over Ethernet). Preporučeno napajanje je 5V jakosti električne struje od 3A. Ako je jakost električne struje znatno manja od 3A dolazi do sporijeg rada računala kao i restartiranja zbog nedostatka električne energije da se obave procesi koji su u tijeku.

Prikaz funkcionalnosti pinova Raspberry Pi Model B računala dan je na sljedećoj slici.



Slika 4.7 Prikaz funkcionalnosti pinova računala Raspberry Pi 4 Model 4 [7]

Iz naziva GPIO pinova vidljivo je da ovo računalo podržava UART, SPI te I2C komunikacijske protokole pomoću kojih možemo komunicirati s odabranim mikrokontrolerom Teensy 3.5 koji također podržava navedene protokole. Radna temperatura okruženja ovog računala je od 0°C-50°C. Ukoliko dolazi do pregrijavanja procesora zbog povišene temperature okoline ili pak zbog obavljanja iznimno složenih softversko-hardverskih radnji, preporuka je na procesor staviti hladnjak ili ventilator ili ukomponirati obje solucije.

U nastavku ćemo objasniti koje radnje je potrebno obaviti kako bismo Raspberry Pi računalo uspješno pokrenuli kao i programsko okruženje koje ćemo koristiti za programiranje.

4.2.1 Prvo pokretanje računala Raspberry Pi model 4 B

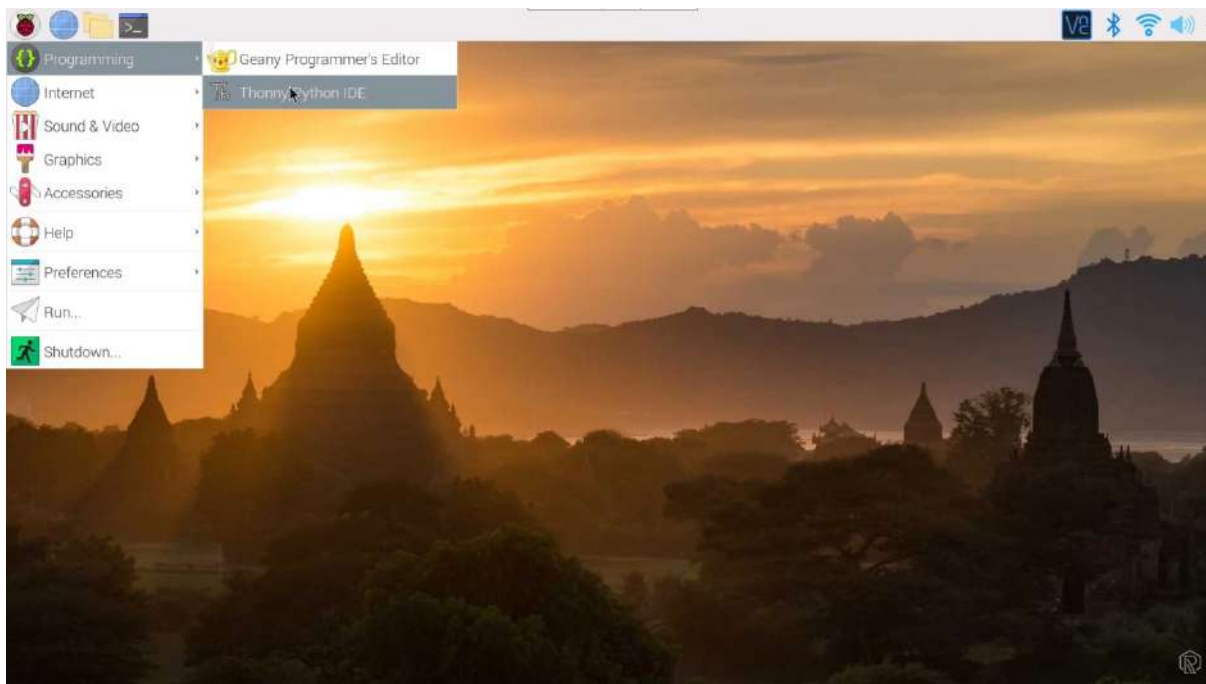
Kako bi se prvi put uspješno pokrenulo Raspberry Pi računalo, mora se instalirati jedan od ponuđenih operativnih sustava. U ovom projektu koristi se službeni operativni sustav Raspberry Pi platforme koji nosi naziv Raspbian i koji se može instalirati na dva načina ponuđena na službenoj stranici Raspberry Pi platforme. Prvi način je instaliranje operativnog sustava korištenjem NOOBS-a „open source“ bootloader-a ili pak koristeći službeni način instalacije korištenjem Raspberry Pi Imager sustava. Za projekt odabrana je instalacija operativnog sustava pomoću NOOBS-a, iako se korištenjem Raspberry Pi Imager sustava instalira isti operativni sustav-Raspbian. Ovaj operativni sustav dolazi u standardnoj verziji koja zauzima približno 1,3 GB memorije dok Raspbian Lite verzija zauzima oko 300 MB prostora što je cijeli gigabajt prostora manje u odnosu na standardnu verziju operativnog sustava što sa sobom povlači i slabije mogućnosti, podržavanje manjeg broja modula i sl.

Za instalaciju operativnog sustava potrebna je memorijska micro SD kartica minimalnog preporučenog kapaciteta od 8 GB do maksimalno podržanog kapaciteta od 32 GB. Uz memorijsku karticu, potrebno nam je i drugo računalo na koje se sa službene Raspberry Pi stranice preuzme NOOBS, te se NOOBS potom prebaci na memorijsku karticu, koja mora biti formatirana. Memorijska kartica na računalu se očitava pomoću čitača kartica koji može biti integriran u dato računalo ili pak kao zaseban adapter kojeg možemo očitavati preko USB porta.

Kada je NOOBS prebačen na micro SD karticu, ista micro SD kartica se postavi u micro SD utor Raspberry Pi računala. Potom se na USB portove Raspberry Pi računala priključe tipkovnica i miš, dok se na micro HDMI izlaz priključi ekran, preko micro HDMI-HDMI kabela kako bi bila moguća ulazno-izlazna interakcija s računalom. Nakon priključivanja miša, tipkovnice i ekrana, Raspberry Pi računalo priključi se, preko USB tip C energetskog porta, na odgovarajući izvor električne energije, istosmjernog napona od 5 V, te jakosti struje od preporučeno 3A, dok ovaj model Raspberry računala sasvim pristojno radi i na jakosti električne struje od 2 A, što je standard za mnoge brze punjače mobilnih uređaja.

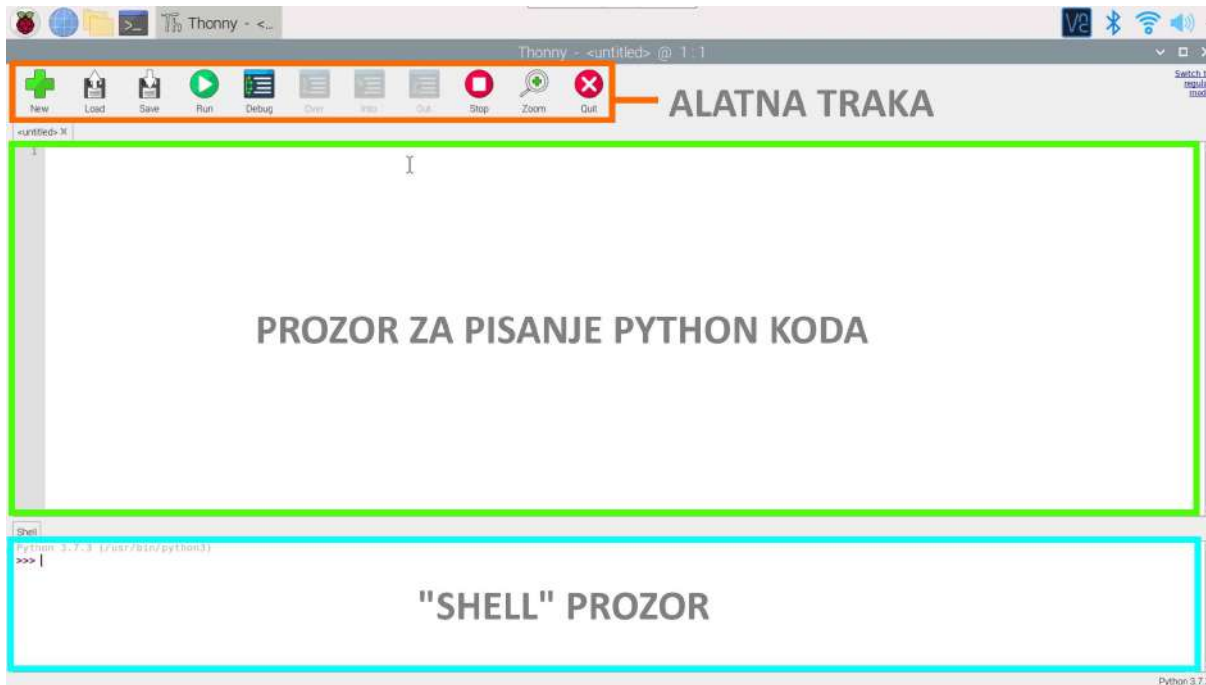
Na ekranu se pojavljuje opcija za odabir operativnog sustava koji se želi instalirati, što je u ovom slučaju Raspbian, te je potrebno odabrati opciju za instalaciju, nakon čega pratimo proces instalacije. Kada je instalacija uspješno okončana, računalo se ponovno pokreće te prikazuje radnu površinu računala.

Budući da Raspberry Pi podržava programski jezik Python, s Raspbianom dolazi automatski instalirano programsko okruženje u koje se može pisati Python programski kod, a naziva se Thonny Python IDE, a može ga se pronaći na način da se mišem klikne na početni izbornik, potom na opciju „Programming“, potom na program Thonny Python IDE, u kojem se piše Python kod na Raspberry Pi računalu.



Slika 4.8 Prikaz odabira programa Thonny Python IDE

Prikaz sučelja Thonny Python IDE programskog okruženja dan je na narednoj slici.



Slika 4.9 Prikaz sučelja programa Thonny Python IDE

Program se upisuje u prozor koji je za to predviđen ili pak, ako želimo koristiti gotov programski kod, možemo ga učitati korištenjem opcije „Load“ iz alatne trake sučelja. Kod se pokreće aktivacijom opcije „Run“, a zaustavljanje pokrenutog koda vrši se aktiviranjem opcije „Stop“. Svi željeni ispisi programa, kao i željene dinamičke interakcije sa napisanim programom odvijaju se u „Shell“ prozoru.

4.2.2 Komandni terminal i instalacija potrebnih modula

Sve potrebne instalacije modula izvode se preko komandnog terminala koristeći „pip“ modul. Pip modul omogućava instalaciju gotovo svakog modula za python programsko okruženje, a prednost u odnosu na uobičajeni način instalacija modula i aplikacija je izostanak brige o tome gdje preuzeti određeni modul, gdje spremiti određeni modul što zna biti vrlo zamoran postupak. Na Internet stranici „PyPi.org“ moguće je pronaći željeni modul čiji se naziv uz ključnu riječ pip ili pip3 upiše u komadni terminal Raspberry Pi računala, nakon čega dolazi do pokretanja instalacija. Kako bismo mogli koristiti pip modul, potrebno ga je prethodno instalirati na način da se u komandnom prozoru upiše naredba „sudo apt-get install python-pip“, te na tipkovnici pritisne tipku „Enter“.

Budući da se određeni moduli „unaprjeđuju“, postoji mogućnost da novija verzija nekog modula neće podržavati određene nazive funkcija i funkcionalnosti iz prethodnih verzija, stoga ako se koristi više modula u kombinacija, iznimno je važno znati verzije određenih modula kako bi izbjegli nekompatibilnosti koje dolaze s novijim verzijama modula.

Nazivi modula i verzije korištene u projektu navedene su u nastavku.

picamera: verzija = 1.13

opencv-python: verzija = 3.4.6.27

Pillow: verzija = 5.4.1

PyAutoGUI: verzija = 0.9.48

numpy: verzija = 1.16.2

pyserial: verzija = 3.4

pytesseract: verzija = 0.3.2

RPi.GPIO: verzija = 0.7.0

Moduli naziva „tkinter“, „math“ i „time“ se koriste u projektu, ali ne dolaze s atributom verzije koja za ove module u konačnici nije ni bitna.

Objašnjenja u koju se svrhu koriste dani moduli navedena su u programskom kodu projekta koje nalazi na referenci [30]

4.3 Hardverske komponente robota

Hardverske komponente korištene na robotu navedene su u nastavku.

- 1 x Baterija
- 4 x Električni motor
- 2 x Kontroleri motora
- 6 x IR Senzor udaljenosti
- 1 x PDB (eng. Power Distributon Board)
- 1 x Razvod žica
- 1 x Žiroskop
- 1 x Senzor Boje
- 2 x Termalna kamera
- 2 x Raspberry Pi optička kamera
- 1 x Multi-kamera adapter
- 1 x Zvučnik
- 1 x Ledica (LED)
- 1 x Solenoid
- 1 x Teensy 3.5
- 1 x Adapter za Teensy 3.5
- 1 x Raspberry Pi Model 4 B
- 1 x USB tip A -micro USB tip B kabel
- 1 x USB tip A -USB tip C kabel
-

Prvi cilj je realizirati kretanje robota. Komponente koje su potrebne za ispunjavanje ovog cilja ukratko su objašnjene su u nastavku.

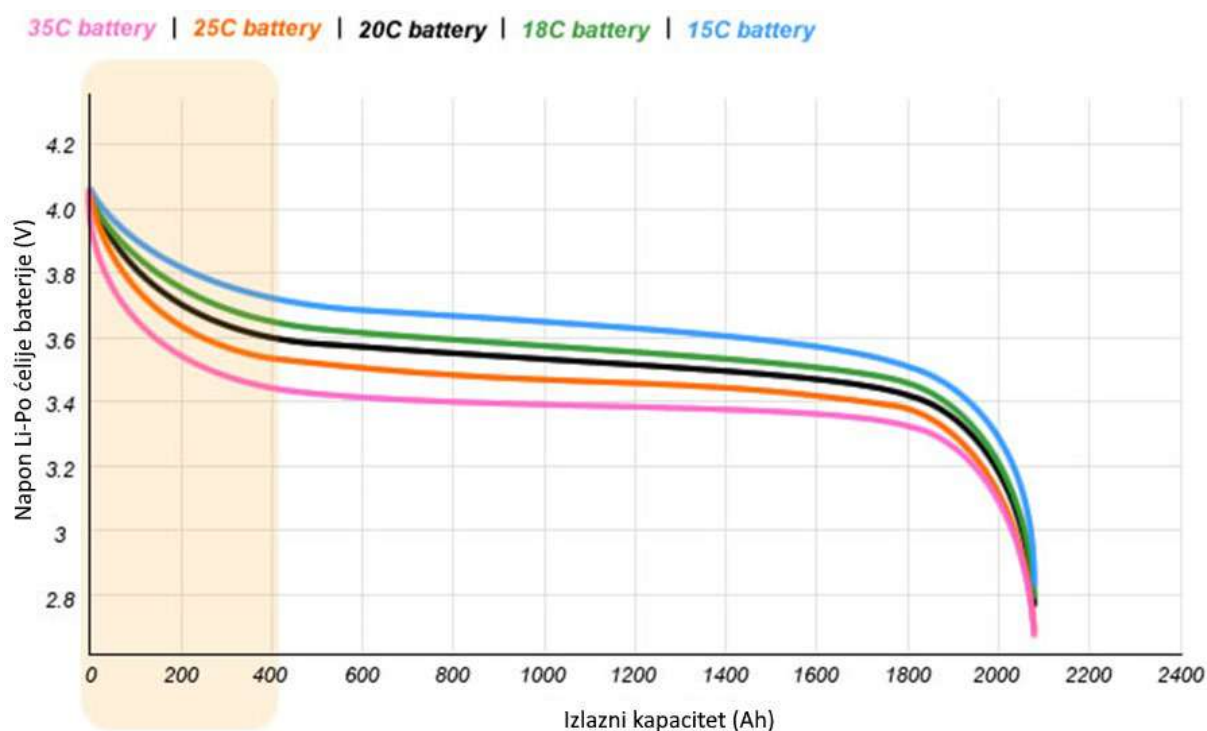
4.3.1 Baterija

Baterija koja se koristi u projektu je Turnigy, nano-tech, 1800 mAh, 25-50C, 3S, Li-Po baterija. Kratko objašnjenje oznaka baterije dano je u nastavku.

Naznaka 1800 mAh označava da je električni kapacitet baterije takav da može davati jakost električne struje 1800 mA ili 1.8 A u vremenskom trajanju od jedan sat. Ako je potrebna jakost električne struje od npr. 0.9 A, ova baterija će toliku jakost struje moći davati dva sata odnosno dvostruko duže vremena jer je ova jakost električne struje dvostruko manja od one koja je definirana u standardnom obliku za vremenski interval od jednog sata. Sada se postavlja pitanje što ako sustav treba višestruko veću jakost električne struje od 1,8 A? Odgovor je relativno jednostavan. Oznaka „C“ predstavlja takozvani „C-rating“ odnosno jakost struje kojom se baterija puni u preporučenim uvjetima, odnosno za bateriju kapaciteta 1800 mAh „idealna“ jakost električne struje kojom bi punili ovu bateriju je upravo 1800 mA, dok je „idealna“ jakost električne struje kojom bi se punila bateriju od npr. 5000 mAh, 5000 mA. Većina baterija omogućava pražnjenje električnom strujom kojom se i puni, dok velik broj baterija omogućava

pražnjenje baterija do približno tri puta većom jakosti od jakosti električne struje punjenja, odnosno podržavaju pražnjenje baterije jakostima električne struje do vrijednosti 3C. Ukoliko je potrebna višestruko veća jakost električne struje na izlazu iz baterije, baterija treba moći podržavati određenu jakost. Za korištenu bateriju C-rating iznosi 25-50C. Prvi broj označava maksimalnu kontinuiranu jakost električne struje kojom se može prazniti baterija, dok drugi broj označava maksimalnu jakost električne struje kojom se može prazniti bateriju jako kratko vrijeme. To vrijeme definira sam proizvođač, a tipično trajanje ovakvog pražnjenja je nekoliko sekundi. Konkretno to znači da se ova bateriju može kontinuirano prazniti jakošću električne struje od $25 \times 1800 \text{ mA}$ što iznosi $45\,000 \text{ mA}$ odnosno 45 A , dok se na trenutke može prazniti jakošću električne struje od $50 \times 1800 \text{ mA}$ što iznosi 90 A što je uistinu impresivna mogućnost. Također je važno napomenuti da veći C-rating uglavnom znači i veći cijenu baterije.

Standardni napon jedne ćelije baterije označava se slovom S, a on je za Li-Po tehnologiju u intervalu od 3.2 V kada je ćelija potpuno prazna do 4.2 V kada je Li-Po ćelija potpuno napunjena. Oznaka 3S znači da je minimalni napon baterije jednak $3 \times 3.2 \text{ V}$ što predstavlja 9.6 V , dok je napon potpuno napunjene baterije 12.6 V . Ako se bateriju ne planira koristiti duži vremenski period, potrebnu ju je dovesti na napon od oko 3.7 V , kako ne bi došlo do gubitka kapaciteta baterije, zbog same kemijske strukture baterije, koja nije predmet ovog diplomskog rada. Napon od 3.7 V odabire se jer je upravo na tome naponu krivulja pražnjenja Li-Po baterije najstabilnija, što svakako nije slučaj za napone od 3.2 V i 4.2 V . Krivulja pražnjenja nasumično odabrane Li-Po baterije prikazana je na narednoj slici.



Slika 4.10 Prikaz izlazne karakteristike nasumično odabrane Li-Po baterije [8]

Iz oblika krivulje pražnjenja Li-Po ćelije koji je prikazan na prethodnoj slici, vidi se da krivulja pražnjenja baterije ima gotovo zanemarive padove napona za vrlo širok raspon izlaznog

električnog kapaciteta, što govori da je baterija na naponu oko 3.7 V u najstabilnijem području očuvanja napona, koji je ključan za očuvanje kemijskih svojstava baterije. Izgled baterije korištene za realizaciju robota je prikazan na narednoj slici.





Slika 4.11 Prikaz baterije korištene u projektu [9]

4.3.2 Električni motori

Za kretanje robota izvršne komponente su četiri istosmjerna motora s odgovarajućim prijenosom na čiju su izlaznu osovinu pričvršćeni odgovarajući kotači.

Istosmjerni motor korišten na projektu je FK-130SH-14225 DC/12V 171111, motor koji dolazi s dugotrajnim četkicama koje su izrađene od karbonskih vlakana. Budući da je brzina vrtnje izlazne osovine ovog motora jako velika te nije upotrebljiva za projekt, ovom motoru potrebno je na izlaznu osovinu ugraditi odgovarajući prijenos, kako bi se dobila odgovarajuća brzina vrtnje izlazne osovine, kao i odgovarajući zakretni moment. Za ovaj projekt, koristi se prijenos omjera 156:1, što znači da se izlazna osovina DC motora okrene 156 puta za 360° dok se izlazna osovina prijenosa okrene jedan put za 360°. Također u projektu se koristi konfiguracija motora koji ima dvije izlazne osovine, od koji jedna strana osovine pokreće mehanizam prijenosa, dok druga strana izlazne osovine DC motora može služiti za ugradnju enkodera kako bi se mogla mjeriti udaljenost koju robot prelazi. Ovu kombinaciju DC motora i prijenosa moguće je pronaći na službenim stranicama kompanije Pololu. Rezultat ove kombinacije DC motora i prijenosa je taj da se izlazna osovina prijenosa motora na naponu od 12V, bez tereta na izlaznoj osovini, vrti 89 puta po minuti što je prihvatljivo za konkretnu primjenu. Karakteristike ove i različitih konfiguracija DC motora i prijenosa prikazane su u narednoj tablici.

Tablica 4.4 Prikaz izlaznih karakteristika određene kombinacije DC motora i prijenosa [10]

Rated Voltage	Brush Type	Stall Current	No-Load Current	No-Load Speed (RPM)	Extrapolated Stall Torque		Max Power (W)	 Single-Shaft (Gearbox Only)	 Dual-Shaft (Gearbox & Motor)
					(kg · cm)	(oz · in)			
12 V	long-life carbon (CB)	1.6 A	0.08 A	550	1.9	26	2.6	25:1 12V CB	25:1 12V CB dual shaft
				450	2.4	33	2.8	31:1 12V CB	31:1 12V CB dual shaft
				220	4.6	64	2.6	63:1 12V CB	63:1 12V CB dual shaft
				180	5.4	75	2.5	78:1 12V CB	78:1 12V CB dual shaft
				140	6.1	85	2.2	100:1 12V CB	100:1 12V CB dual shaft
				110	7.8	110	2.2	125:1 12V CB	125:1 12V CB dual shaft
				89	9.0	130	2.0	156:1 12V CB	156:1 12V CB dual shaft
				72	10	140	1.9	195:1 12V CB	195:1 12V CB dual shaft
				55	14	190	2.0	250:1 12V CB	250:1 12V CB dual shaft
				44	16	220	1.8	313:1 12V CB	313:1 12V CB dual shaft
				36	21	290	1.7	391:1 12V CB	391:1 12V CB dual shaft
				28	25	350	1.5	488:1 12V CB	488:1 12V CB dual shaft

Prikaz odabrane kombinacije DC motora i prijenosa, bez i sa enkoderom prikazana je na sljedećoj slici.

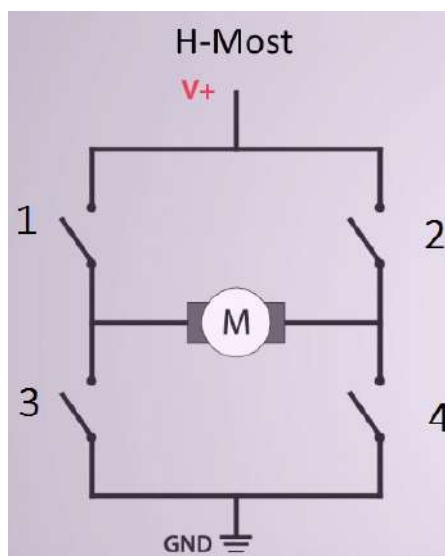


Slika 4.12 Prikaz odabrane kombinacije DC motora i prijenosa sa i bez montiranog enkodera [10]

4.3.3 Kontroler motora

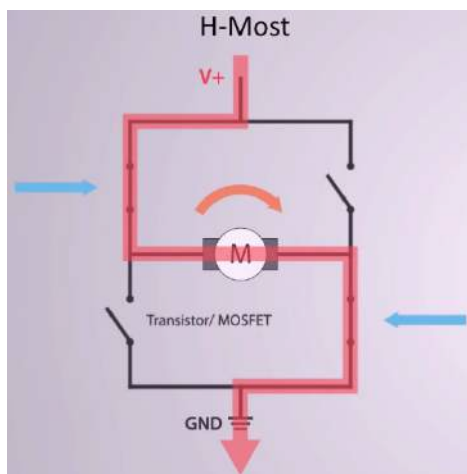
Nakon ispravno odabranog odgovarajućeg spremnika električne energije tj. baterije, istosmjernih motora, te mikrokontrolera pomoću kojeg se upravlja istosmjernim motorima, pojavljuje se prepreka vezana za način povezivanja motora s odgovarajućim napajanjem i mikrokontrolerom. Naime navedeni DC motori, što se vidi u prethodnoj tablici, vuku jakost električne struje u rasponu od 0.08 A do 1.6 A, na 12 V. Kako mikrokontroler Teensy 3.5 na svojim pinovima ne može dati veći napon od 5 V i veću jakost električne struje od 10 mA odnosno 0.001A, potrebno je ukomponirati dodatnu komponentu koje ima mogućnost

očitavanja PWM signala pomoću kojeg želimo upravljati DC motore, kao i mogućnost da DC motorima daje jakost struje u intervalu od 0.08 A do 1,6 A te napon od 0 V do 12 V. Komponenta koje omogućuje navedeno naziva se kontroler motora koji uz sve navedeno može i mijenjati smjer vrtnje motora što je bitno jer tako robot može ići i unatrag što je jedan od zahtjeva projekta. Mijenjanje smjera motora realizira se pomoću sklopki, u većini slučajeva realiziranih pomoću tranzistora, koje mijenjaju smjer struje koja protječe kroz DC motor te samim tim i smjer vrtnje izlazne osovine motora. Komponenta kontrolera koja je zadužena za mijenjanje smjera vrtnje motora naziva se H-most. Konstrukcija i princip promjene smjera struje motora prikazan je na sljedećim slikama. Kada su sve sklopke otvorene struja ne prolazi kroz DC motor.



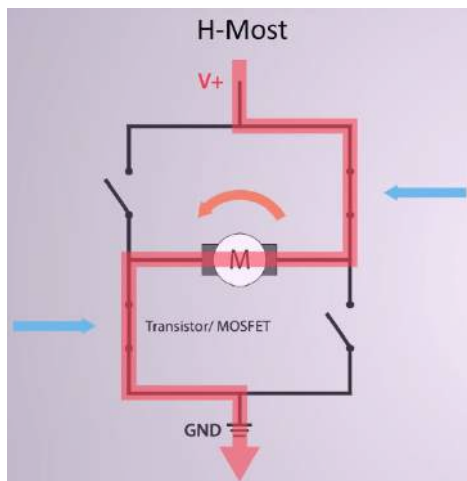
Slika 4.13 Prikaz H-mosta, sve stezaljke otvorene [2]

Kada se zatvore sklopke 1 i 4 električna struje prolazi kroz DC motora u smjeru prikazanom na sljedećoj slici.



Slika 4.14 H-most, stezaljke 1 i 4 zatvorene [2]

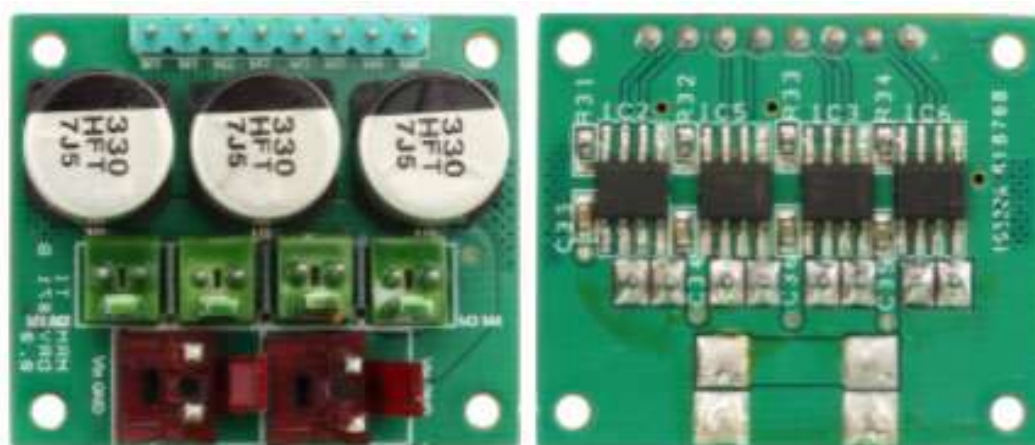
Ukoliko se zatvore sklopke 2 i 3, a otvorenim adrže sklopke 1 i 4 električna struja prolazi kroz DC motor u smjeru prikazanom na sljedećoj slici.



Slika 4.15 H-most, stezaljke 2 i 3 zatvorene [2]

Uz H-most kontroler motora posjeduje pinove koji imaju sposobnost očitavanja PWM signala te ovisno o iznosu i predznaku PWM signala kojeg šalje mikrokontroler Teensy 3.5, kroz H-most se šalje određena jakost električne struje, u određenom smjeru, koja dolazi iz stezaljki napajanja kontrolera motora, te se na taj način može kontrolirati brzina i smjer vrtnje motora odnosno brzina kretanja i rotiranja robota.

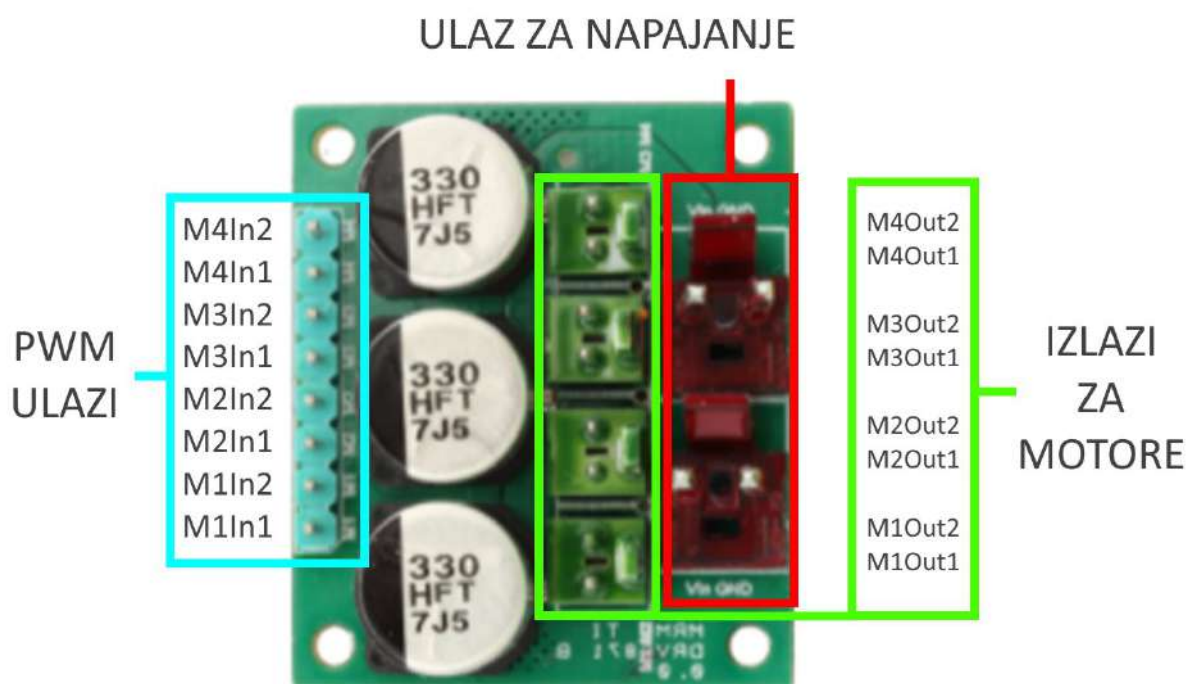
Kontroler motora koji se koristi u projektu, prikazan je na narednoj slici.



Slika 4.16 Prikaz kontrolera motora korištenog u projektu [11]

Ovaj kontroler može kontrolirati 4 DC motora s četkicama, što je vidljivo iz sljedeće slike na kojima su svijetlo plavom bojom naznačena po dva PWM ulaza za određeni DC motor, od kojih

jedan PWM ulaz služi za određivanje brzine vrtnje motora dok drugi PWM ulaz služi za određivanje smjera vrtnje motora. Uz PWM ulaze na ovom kontroleru nalaze se i ulazi za napajanje motora naponima od 6.5 V - 45 V koji su označeni oznakama Vin i GND, te četiri izlaza na koje se spajaju motori, od kojih svaki izlaz zasebno može davati jakost električne struje od 3.6 A što je i više nego dovoljno za potrebe projekta, imajući u vidu da je maksimalna jakost električne struje koju može koristiti odabrani DC motor tek 1.6 A.



Slika 4.17 Prikaz funkcionalnosti pinova kontrolera motora [11]

Ako se za određeni projekt koriste motori bez četkica, tada je potrebno koristiti kontroler motora koji je namijenjen za izvedbe motora koji rade bez četkica (eng. Brushless Motor).

4.3.4 Senzori udaljenosti

Kada je realizirano kretanje robota potrebno je omogućiti robotu da „vidi“ svoje okruženje. Prva mogućnost koju želimo da Raspberry Pi -Teensy labirint robot spasilac posjeduje je ta da vidi prepreke odnosno zidove labirinta koje ga okružuju. Za ovu svrhu koristit ćemo senzore udaljenosti koji mjere udaljenost robota od zida te ukoliko je npr. zid previše blizu robota potrebno je poduzeti određenu radnju. Ultrazvučne senzore udaljenosti ne želimo koristiti iz razloga što dolazi do višestrukog reflektiranja ultrazvučnog vala u raznim smjerovima od zidove labirinta te dolazi do zbunjivanja robota na način da se vrlo lako može dogoditi da robot očita višestruko reflektirani val koji samim tim što je višestruko reflektiran u za nas nepoznatom smjeru ne nosi nam nikakvu značajnu informaciju, odnosno stvara smetnju. Da bi spriječili višestruku refleksiju valova na čiji se izvor oslanjamo potrebno je koristiti senzore koji koriste

drugi princip rada. Kao najbolje rješenje su se nametnuli LIDAR senzori (eng. Light Detection And Ranging) iz razloga što ovi senzori koriste lasersku svjetlost te su samim tim precizniji, brži i otporniji na promjene trenutnih uvjeta okoliša, iz razloga što laserskoj svjetlosti nije potreban medij za širenje, dok ultrazvuku koji je mehanički val, svojstva ovise o trenutnim uvjetima medija kojim se širi, koju mogu biti vrlo nestabilni. Kada smo odabrali principi rada senzora udaljenosti kojeg koristimo u projektu potrebno je odabrati proizvođača i model senzora koji ćemo koristiti vodeći računa da odabrani senzor posjeduje adekvatan mjerni opseg, kompatibilno sučelje za očitavanje vrijednosti udaljenosti pomoću mikrokontrolera koji je odabran, kao i kompaktne dimenzije.

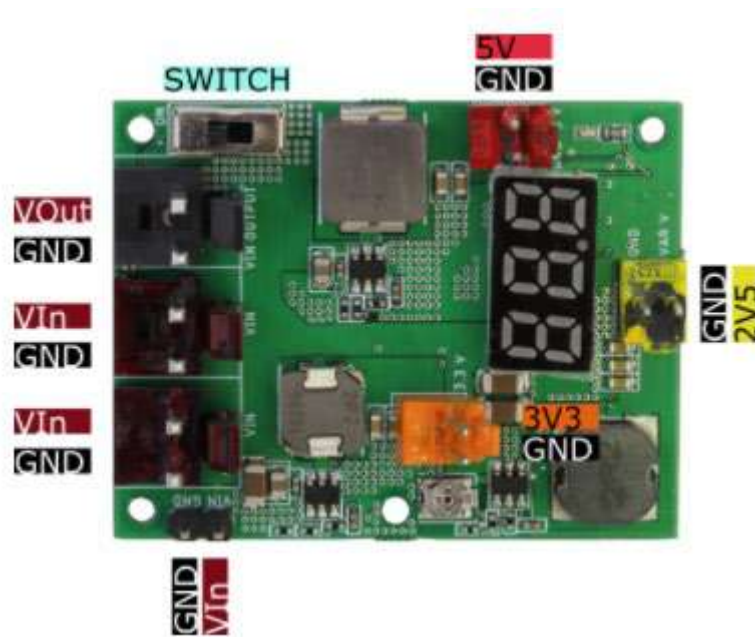
U projektu su korišteni MRMS LIDAR 2m VL53L0X koji su se loše pokazali te su zamijenjeni Sharp GP2Y0A41SK0F senzorima.



Slika 4.18 Prikaz Sharp GP2Y0A41SK0F senzora s funkcionalnostima pinova [12]

4.3.5 PDB (eng. Power Distribution Board)

Budući da će u projektu trebati više naponskih razina istovremeno, 12 V za DC motore, 5 V i 3.3 V za mnogobrojne senzore koje koristimo, a koji će biti navedeni i objašnjeni u nastavku, kao i „velike“ jakosti električne struje, potrebno je koristiti određeni transformator odnosno PDB pločicu (eng. Power Distribution Board) kako bi svim željenim električnim izlazima imali jednostavniji pristup. PDB koji smo odabrali je MRMS Power Supply 3x C, voltmeter (mrm-ps3xc) koji je prikazan na narednoj slici.

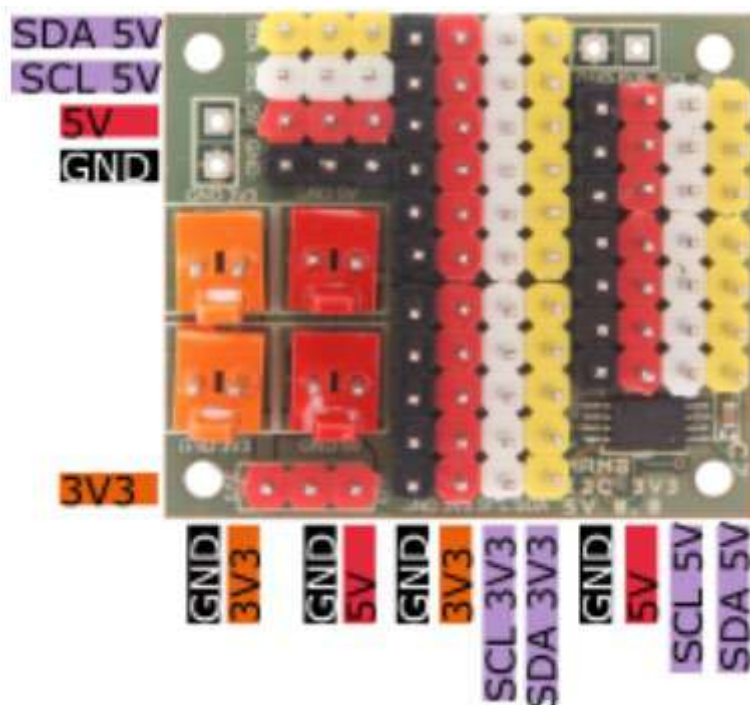


Slika 4.19 Prikaz PDB pločice s funkcionalnostima pinova [11]

Ovaj PDB posjeduje 3 odnosno 4 izlaza ako računamo izlaz na prethodnoj slici označen kao VOut koji prosljeđuje stanja s ulaza Vin izvoda. Izlaz od 5 V omogućuje kontinuiranu izlaznu jakost električne struje kao izlaz od 3,3 V, dok varijabilni izlaz na prethodnoj slici označen oznakom 2V5 posjeduje trimer pomoću kojeg izlazni napon na ovome izlazu možemo podesiti na željenu vrijednost iz intervala do 0,76 V do 7 V, dok je izlazna jakost električne struje na ovom izlazu 3 A. Izlaz od 5 V ćemo koristiti za napajanje Raspberry Pi računala budući da ono zahtjeva kontinuiran napon od 5 V kao i struju od oko 3 A, dok ćemo izlaz označen oznakom 3V3 koristiti za napajanje određenog dijela senzora. VOut izlaz koristimo za napajanje kontrolera motora. Na bilo koji od dva Vin ulaza direktno se spaja baterija preko odgovarajućeg konektora, jer upravo preko Vin pinova vršimo napajanje ove PDB pločice. Pri odabiru PDB pločice uz mogućnosti izlaza, važno je obratiti pažnju na podatak koji govori kolike su mogućnosti ulaznih pinova napajanja, odnosno za koji je naponski interval određeni PDB namijenjen, kako bi odabrani PDB bio kompatibilan s odabiranom baterijom. Vin ulazi ove PDB pločice omogućuju dovođenje napona u intervalu od 6,5 V do 16 V u koji spada i 3S konfiguracija baterije koja je odabrana za ovaj projekt, čiji je maksimalni napon 12.6 V a minimalni napon 9.6 V. Uz navedene mogućnosti ulaza i izlaza ova PDB pločica posjeduje integrirani voltmetar pomoću kojeg mjeri ulazni napon baterije te ga prikazuje na 7-segmentnom LED ekranu koji je također integriran u samu pločicu, kao i sklopka pomoću koje se omogućuje ili prekida dotok električne energije iz baterije. Jednostavnije rečeno pomoću ove sklopke vršit će se paljenje i gašenje cjelokupnog sklopovlja robota.

4.3.6 Razvod žica

Kada smo dobili željene naponske razine, potrebno je povećati broj izlaza određenih naponskih razina, kako bi na njih mogli spojiti više uređaja. Za primjer uzmimo pretpostavku da će u projektu potrebno biti deset senzora kojima je napon napajanja 3,3 V, a mi posjedujemo samo jedan izlaz od 3,3V na PDB pločici, i nedovoljan broj pinova koji omogućuju napajanje naponom od 3,3 V na mikrokontroleru i računalu robota. Osnovna ideja je paralelno spojiti određeni broj pinova na određeni pin koji omogućuje napajanje naponom 3,3 V. Sada se postavlja pitanje koji 3,3 V pin odabrati, pin s računala, pin s mikrokontroler ili pak pin s PDB pločice. Budući da se radi o paralelnom električnom spoju, napon pinova će biti konstantan, dok će se spajanjem svakog novog uređaja na pinove koji su dodani u paralelni spoj s određenim 3,3V pinom povećavati potrebna jakost električne struje koju 3,3V može dati, stoga je najbolja opcija odabrati pin s PDB pločice koji na svom izlazu može dati jakost električne struje od 3 A odnosno 3000 mA, što je višestruko više od 3,3V pina Teensy 3.5 mikrokontrolera koji na svom izlazu daje jakost od 250mA ili pak od Raspberry Pi 3,3V pina koji ovisno o modelu na svom izlazu može dati jakost električne struje u rasponu od 50-500 mA. Nakon što je donesena odluka kojem pinu se dodaje paralelni spoj pinova, potrebno je izvršiti paralelno dodavanje pinova. Ovu operacija može se realizirati na način da pinove jednostavno zalemimo na odbrani izlaz ili pak pronađemo modul koji omogućuje realizaciju navedenog cilja. Odabrana je ova druga opcija odnosno modul razvoda žica oznake MRMS I2C 3V3 5V Distribution Pins (mrm-distrib-a) koji je prikazan na sljedećoj slici.



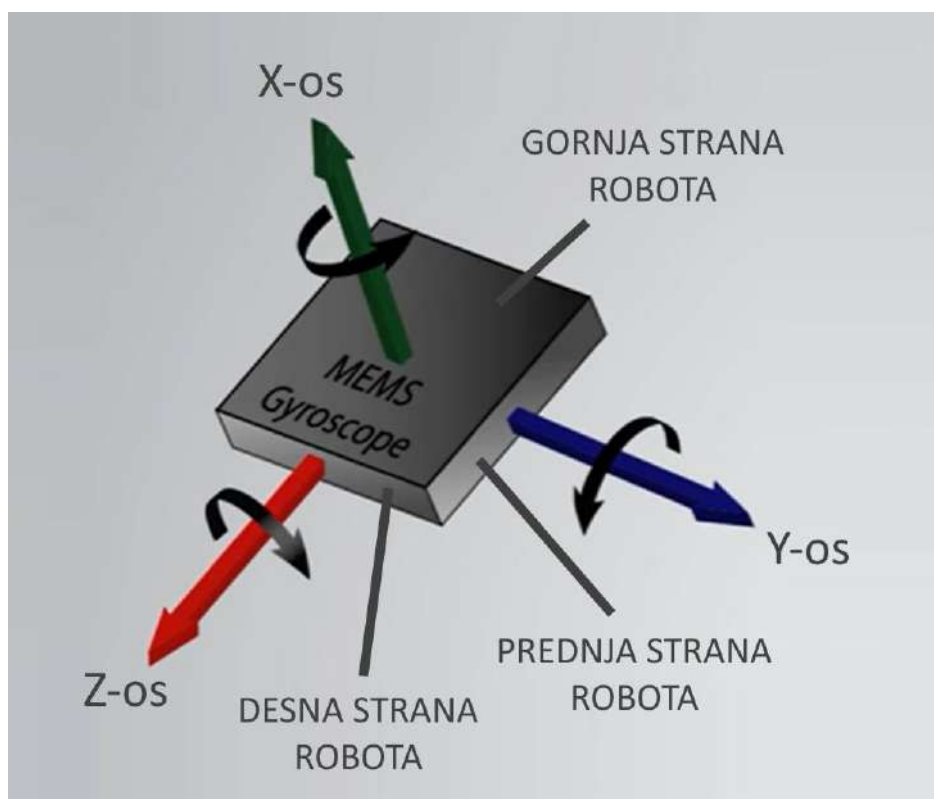
Slika 4.20 Prikaz razvoda žica s funkcionalnostima pinova [11]

Za aktivaciju svih pinova napona 3,3V potrebno je dovesti napajanje od 3,3V na barem jedan od dva ulaza koji su za dati napon predviđeni, a na prethodnoj slici su obojani narančastom bojom.

Za aktivaciju svih pinova napona 5V potrebno je dovesti napajanje od 5V na barem jedan od dva ulaza koji su za dati napon predviđeni, a na prethodnoj slici su obojani crvenom bojom.

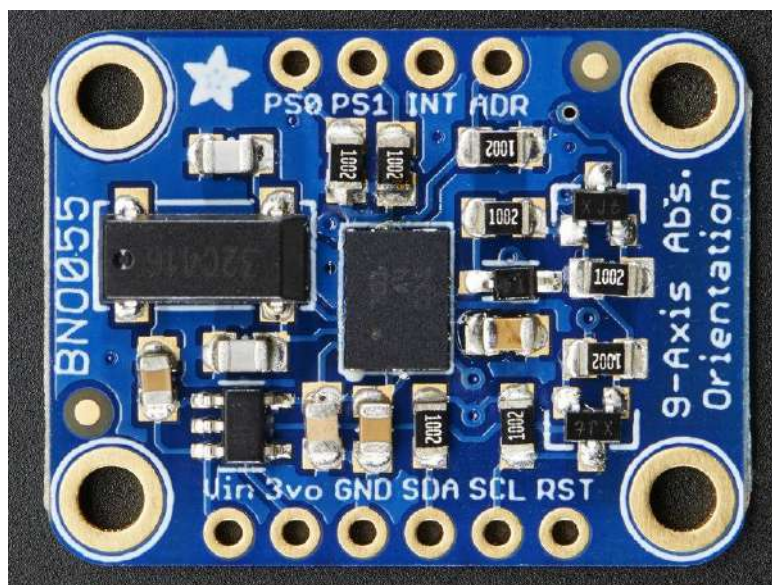
4.3.7 Žiroskop

Budući da želimo da robot raspolaže podacima o svojoj orijentaciji potrebno je koristiti žiroskop. Žiroskop je uređaj koji omogućava očitavanje podatka o orijentaciji određene ravnine. Za ovaj projekt dovoljna mogućnost mjerenja orijentacije na trima osima, odnosno ravninama, koje označavamo X,Y i Z ravninom. X,Y i Z ravnine su prikazane na sljedećoj slici. Za projekt je bitna orijentacija oko X-osi pomoću koje je moguće saznati kreće li se robot pravocrtno, te orijentacija oko Z-osi pomoću koje se može zaključiti da li je robot naišao na određeni nagib.



Slika 4.21 Prikaz osi rotacije robota [13]

Žiroskop koji se koristi u projektu nalazi se u uređaju Adafruit BNO055 koji je prikazan na sljedećoj slici.



Slika 4.22 Prikaz Adafruit BNO055 uređaja [14]

Iz priložene slike vidljivo je da se ovaj senzor sastoji od dvije skupine pinova, odnosno gornje i donje skupine. U nastavku je ukratko objašnjena funkcija pinova prikazanih na prethodnoj slici.

PS0 i PS1-su pinovi koji služe za promjenu načina rada senzora odnosno ovi pinovi mogu biti konfigurirani za rad HID-I2C i UART (eng. Human Interface Device) protokola kao i za ažuriranje senzora. Ovi pinovi uglavnom ostaju nekorisćeni.

INT-ovaj pin omogućuje kreiranje „interrupt“ signala kada se dogodi određeni događaj (senzor detektira kretanje uređaja). Ova mogućnost još nije podržana od strane službene Adafruit biblioteke, ali sklopovlje posjeduje mogućnosti kreiranja pometnutog signala

ADR-pin služi za promjenu I2C adrese ukoliko se želi komunicirati s više uređaja preko I2C protokola. Standardna adresa ovog uređaja je 0x28, a ako na ADR pin dovedemo napon od 3 V tada adresa uređaja postaje 0x29.

VIN-pin služi za dovođenje napona u rasponu od 3,3 V -5 V, koji služe za napajanje sklopovlja
3vo-pin služi kao izlazni pin napajanja od 3,3 V s maksimalnom jakosti električne struje od 50 mA.

GND-pin služi za zatvaranje strujnog kruga napajanja senzora.

SDA-je podatkovni kanal za I2C komunikacijski protokol

SCL-kanal za takti signal za I2C komunikacijski protokol

RST-služi za hardversko restartiranje senzora, restartiranje se odvija na način da se ovaj pin dovede u energetska-logičko stanje niske razine (eng. LOW), a potom u energetska-logičko stanje visoke razine (eng. HIGH). Ovaj pin je tolerantan za napone do 5 V.

Uz navedeni žiroskop unutar Adafruit BNO055 uređaja, se nalaze akcelerometar i magnetometar. Podacima o željenim orijentacijama pristupamo preko SDA i SCL pinova uređaja, odnosno I2C protokola.

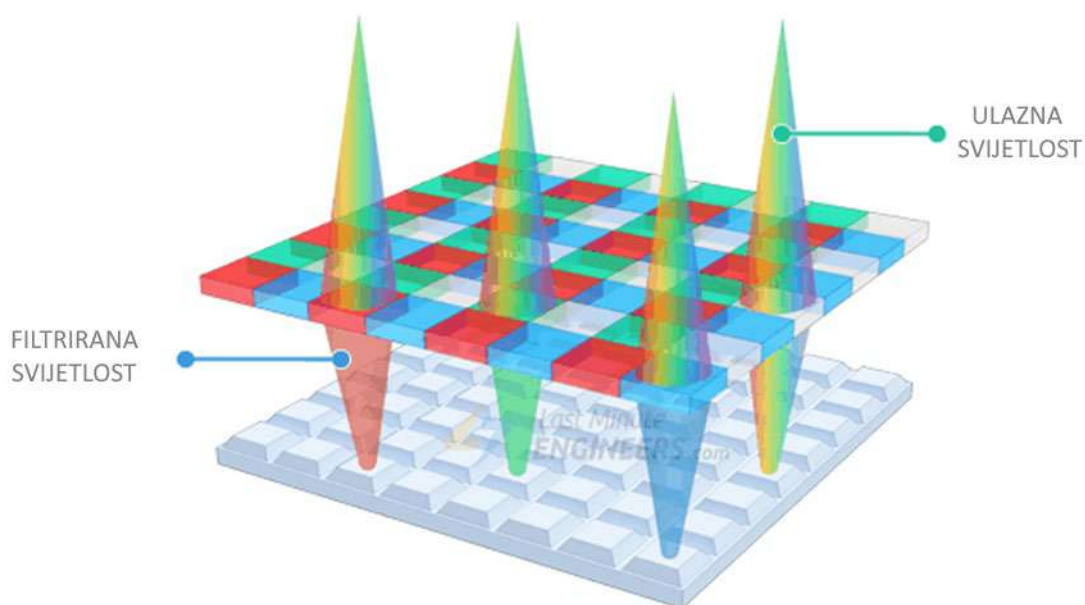
4.3.8 Senzor boje

Sljedeća funkcionalnost koju je potrebno realizirati na robotu je mogućnost prepoznavanje crnih i reflektivno sivih polja labirinata po kojima se robot kreće. Za ovu funkciju odabran je Arduino TCS230/TCS3200 senzor boje koji je prikazan na narednoj slici.



Slika 4.23 Prikaz TCS230/TCS3200 senzora boje [15]

Ovaj senzor sastoji se od 8x8 matrice foto-dioda. Ispred 16 foto-dioda nalazi se filter crvene svjetlosti, ispred sljedećih 16 foto-dioda nalazi se filter zelene svjetlosti, dok se ispred sljedećih 16 foto-dioda nalazi filter plave svjetlosti, ispred posljednjih 16 foto-dioda nema filtera boje. Prikaz 8x8 matrice foto-dioda s i bez određenih filtera dan je u nastavku.



Slika 4.24 Prikaz 8x8 matrice senzora TCS230/TCS3200 [16]

Svih 16 foto-dioda ispred kojih se nalazi filter crvene svijetlosti povezano je paralelno, svih 16 foto-dioda ispred kojih se nalazi filter zelene svijetlosti povezano je paralelno, svih 16 foto-dioda ispred kojih se nalazi filter plave svijetlosti povezano je paralelno dok je i svih 16 foto-dioda ispred kojih se ne nalazi filter boje također povezano paralelno što omogućuje očitavanje svake osnovne boje i intenziteta zasebno, što omogućuje prepoznavanje velikog spektra boja budući da su se sve boje mogu dobiti određenim kombinacijama osnovnih boja odnosno crvene, zelene i plave boje.

Pomoću različitih energetsko-logičkih kombinacija kontrolnih pinova omogućeno je zasebno očitavanja svake od 4 navedene skupine foto-dioda. Prikaz rezultata kombinacija S2 i S3 pinova dan je u nastavku.

Tablica 4.5 Prikaz rezultata kombinacija S2 i S3 pinova senzora TCS230/TCS3200 [16]

S2	S3	Photodiode type
LOW	LOW	Red
LOW	HIGH	Blue
HIGH	LOW	Clear (No filter)
HIGH	HIGH	Green

Vrijednosti električne struje određene kombinacije foto-dioda se pomoću strujno-frekventnog pretvarača koji je ugrađen u sam senzor pretvara u frekvenciju pomoću koje saznajmo o kojoj se boji radi. Izlazna frekvencija je u rasponu od 2 Hz do 500 KHz, a pomoću električno-logičkih kombinacija kontrolnih pinova S0 i S1, moguće je vršiti skaliranje izlazne frekvencije kako bi ovaj senzor bio kompatibilan sa većim brojem mikrokontrolera i ostalih uređaja koji mogu očitavati frekvenciju. Prikaz rezultata kombinacija S0 i S1 pinova dan je u nastavku.

Tablica 4.6 Prikaz rezultata kombinacija S0 i S1 pinova senzora TCS230/TCS3200 [16]

S0	S1	Output frequency scaling
LOW	LOW	Power down
LOW	HIGH	2%
HIGH	LOW	20%
HIGH	HIGH	100%

Iz prethode tablice vidljivo je da je izlaznu frekvenciju moguće skalirati na 2 %, 20% i 100% izvorne izlazne vrijednosti frekvencije senzora. Preostali pinovi ovog senzora objašnjeni su u nastavku.

Vcc-pin služi na napajanje sklopovlja senzora naponima u intervalu od 2,7 V do 5,5 V.

GND-pin služi za zatvaranje strujnog kruga napajanja senzora.

OE-(eng. Output Enabled) je pin koji omogućuje slanje frekvencije na izlaz senzora. U većini slučajeva je predefinirano aktivan, a ukoliko nije aktivira se dovođenjem električno-logičke niže razine na njegov ulaz.

OUT-pin koji daje izlazni pravokutni signal za sklopovlje koje podržava TTL logiku

Optimalna udaljenost senzora od površine kojoj želimo znati boju je u intervalu od 1 cm do 3,5 cm.

Senzor boje potrebno je kalibrirati na način da se na optimalnu udaljenost postavi određeni predmet crne boje te se vrši očitavanje maksimalne vrijednosti R,G i B kanala, potom se na optimalnu udaljenost od senzora postavi predmet bijele boje te vršimo očitavanja minimalne vrijednosti R,G i B kanala. Vrijednosti između očitanih minimalni i maksimalnih vrijednosti R, G i B kanala boje predstavljaju radno područje senzora unutar kojeg je moguće prepoznavati ostale boje koje su potrebne, dok vrijednosti iznad maksimalnih i ispod minimalnih vrijednosti R, G i B kanala boje jednostavno odbacimo.

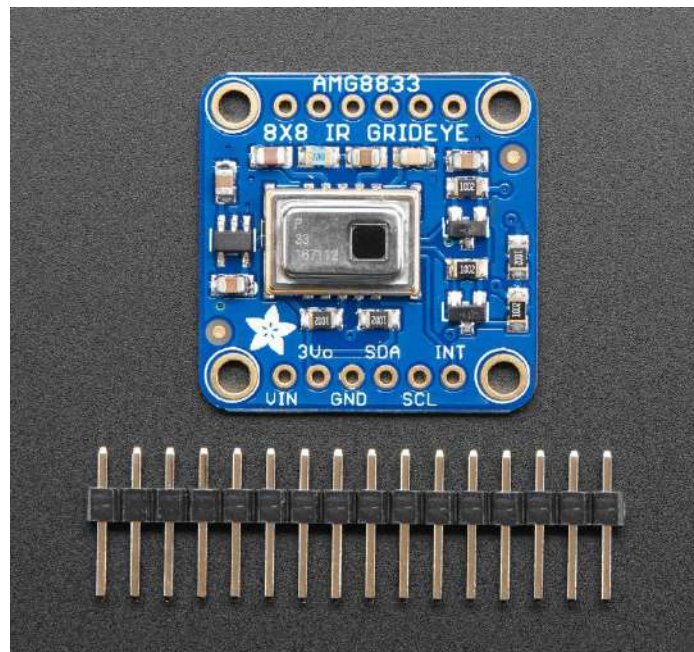
Nakon kalibracije radnog područja senzora, za svaku boju koju želimo da senzor prepozna, ispred senzora na optimalnu udaljenost postavlja se predmet određene boje koju je potrebno

prepoznati te očitati odnosno upisati maksimalne i minimalne vrijednosti R,G i B kanala kao granice intervala između kojih se nalaze vrijednosti koje opisuju željenu boju. Svaka boja posjeduje svoj interval vrijednosti R,G i B kanala što omogućuje da ovisno o očitanim vrijednostima R, G i B kanala možemo znati o kojoj se boji zapravo radi. Za svaku boju potrebno je ponavljati samo postupak kalibracije boja, pratiti minimalne i maksimalne vrijednosti R, G i B kanala kako bi definirali željeni interval za željenu boju.

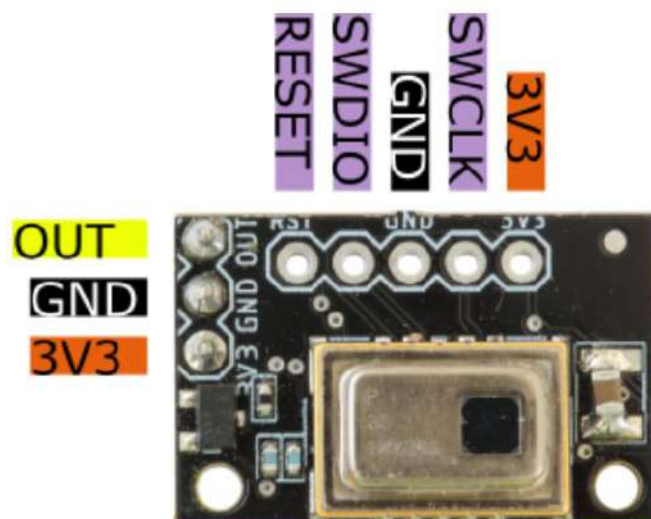
4.3.9 Termalna kamera

Budući da robot treba prepoznati temperaturnu žrtvu, odnosno detektirati izvor topline koji simulira temperaturu čovjeka u temperaturnom intervalu od 28°C do 40°C potrebno je koristiti senzor koji omogućava prethodno navedeno. Uređaj koji omogućuje očitavanje temperature za ovaj projekt je termalna kamera. Termalna kamera za razliku od uobičajenih kamera koje detektiraju reflektiranu svjetlost, detektira stupanj zagrijanost nekog tijela, odnosno temperaturu. Budući da kompaktne dimenzije za robot igraju vrlo važnu ulogu odabrat ćemo termalnu kameru koja radi na principu 8x8 senzorne matrice, odnosno mjeri temperaturu u 64 točke što je za zahtjeve projekta i više nego dovoljno.

Iz razloga što već posjedujemo MRMS Thermal Array Sensor, analog (mrm-therm-b-an) termalnu kameru, u projektu ćemo koristiti nju iako postoje i bolje opcije kao što je npr. Adafruit AMG8833 IR Thermal Camera. Obije kamere prikazane su u nastavku.



Slika 4.25 Prikaz Adafruit AMG8833 IR termalne kamere [17]



Slika 4.26 Prikaz mrm-therm-b-an termalne kamere s funkcionalnostima pinova [11]

Budući da je za projekt odabrana MRMS termalna kamera, ukratko ćemo objasniti način na koji ćemo ju koristiti u projektu. OUT, GND i 3V3 su pinovi koji su korišteni u projektu i nalaze se s lijeve strane na prethodno prikazanoj slici MRMS kamere.

OUT-pin predstavlja pin koji daje izlaznu analognu usrednjenu vrijednost mjerenja temperature u 64 točke objekta koji se nalazi ispred kamere

3V3-pin služi na napajanje sklopovlja senzora naponom od 3,3 V

GND-pin služi za zatvaranje strujnog kruga napajanja senzora.

Cjelokupne karakteristike odabrane termalne kamere dane su u nastavku.

Tablica 4.7 Prikaz karakteristika MRMS termalne kamere dan je u nastavku

Ulazni napon	3.3 V
Broj mjernih točaka	Dvodimenzionalno polje mjernih točaka, 8 x 8 (64 točke)
Najmanja temperatura mjerenja	-20 °C
Najveća temperatura mjerenja	100 °C
Točnost	Tipično ± 2.5 °C. Detektira čovjeka na 7 m i manje (referentna vrijednost)
Kut mjerenja	Tipično 60 °
Struja potrošnje	Tipično 4.5 mA (normalni način rada)
Vrijeme pokretanja	50 ms (vrijeme da se pokrene komunikacija nakon starta)
Sučelje	Analogni izlaz 0-3.3 V pomoću 32-bitnog mikrokontrolera
Frekvencija mjerenja	Tipično 10 mjerenja (od 64 točke)/sec ili 1 mjerenje/sec
Mogućnost dogradnje firmwarea	Da
Zaštita od obrnutog ulaznog napona	Da
Status LED	Da
Standardizirani format	Ne

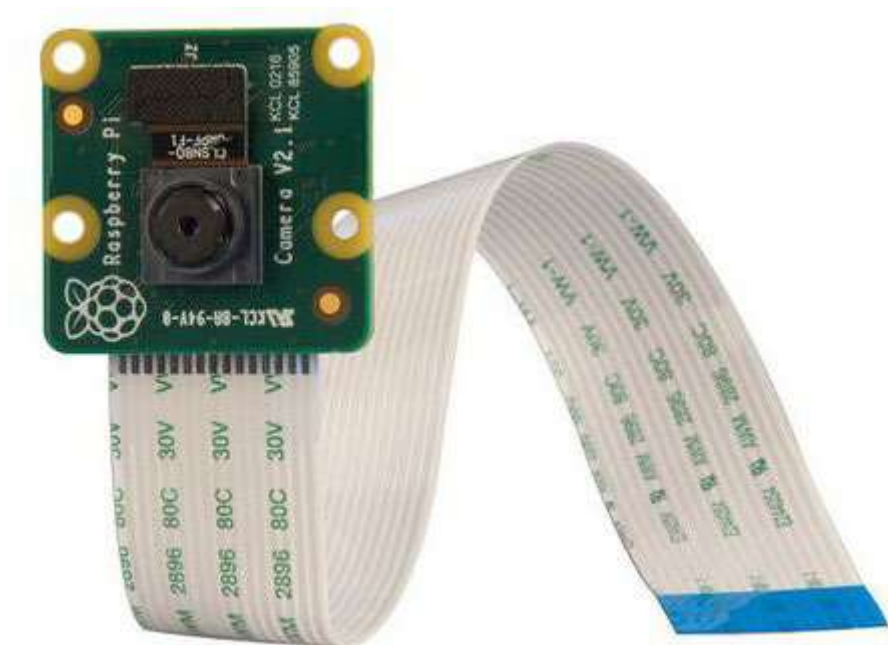
Sve što je potrebno ovom senzoru je dovesti mu napajanje i očitati analognu vrijednost pomoću Teensy 3.5 mikrokontrolera, iz razloga što Raspberry Pi nema mogućnost očitavanja analognih vrijednosti. Druga mogućnost je koristiti poseban ADC pretvarač koji analogni signal termalne kamere prevodi u razumljiv digitalni signal Raspberry Pi računalu. Takav ADC je svakako MCP3008 ADC koji je prikazan na sljedećoj slici.



Slika 4.27 Prikaz MCP3008 analogno-digitalnog pretvarača [18]

4.3.10 Raspberry Pi kamera

Raspberry Pi Camera Module V2 naziv je kamere obarane za ovaj projekt, prikazana je u nastavku skupa s kabelom pomoću kojeg se povezuje na CSI port Raspberry Pi računala.



Slika 4.27 Prikaz Raspberry Pi Kamera Modula V2 [19]

Službene karakteristike kamere u koje nećemo detaljno ulaziti, dane su u nastavku.

Tablica 4.8 Prikaz karakteristika određenih Raspberry Pi kamera [20]

	Camera Module v1	Camera Module v2	HQ Camera
Net price	\$25	\$25	\$50
Size	Around 25 × 24 × 9 mm		38 × 38 × 18.4mm (excluding lens)
Weight	3g	3g	
Still resolution	5 Megapixels	8 Megapixels	12.3 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90	1080p30, 720p60 and 640 × 480p60/90	1080p30, 720p60 and 640 × 480p60/90
Linux integration	V4L2 driver available	V4L2 driver available	V4L2 driver available
C programming API	OpenMAX IL and others available	OpenMAX IL and others available	
Sensor	OmniVision OV5647	Sony IMX219	Sony IMX477
Sensor resolution	2592 × 1944 pixels	3280 × 2464 pixels	4056 × 3040 pixels
Sensor image area	3.76 × 2.74 mm	3.68 × 2.76 mm (4.6 mm diagonal)	6.287mm x 4.712 mm (7.9mm diagonal)
Pixel size	1.4 μm × 1.4 μm	1.12 μm × 1.12 μm	1.55 μm × 1.55 μm
Optical size	1/4"	1/4"	
Full-frame SLR lens equivalent	35 mm		
S/N ratio	36 dB		
Dynamic range	67 dB @ 8x gain		
Sensitivity	680 mV/lux-sec		
Dark current	16 mV/sec @ 60 C		
Well capacity	4.3 Ke-		
Fixed focus	1 m to infinity		N/A
Focal length	3.60 mm +/- 0.01	3.04 mm	Depends on lens
Horizontal field of view	53.50 +/- 0.13 degrees	62.2 degrees	Depends on lens
Vertical field of view	41.41 +/- 0.11 degrees	48.8 degrees	Depends on lens
Focal ratio (F-Stop)	2.9	2.0	Depends on lens

Od navedenih karakteristika pažnju je potrebno obratiti na maksimalan broj slika u sekundi koje ova kamera može hvatati u 1080 rezoluciji. Za ovaj projekt 30 FPS (eng. Frames Per Second) je i više nego dovoljno, kao i rezolucija kamere od 8 megapikela. Promjenjiva žarišna dužina leće kao i prilagodljiv kut gledanja kamere (eng. FOV-Field Of View) svakako bi dobro došli, ali budući da promjenjive navedene veličine kamere sa sobom donose i veće dimenzije, zbog jednostavnosti konstrukcije tijela robota odlučili smo se za Camera Module v2 umjesto HQ kamere.

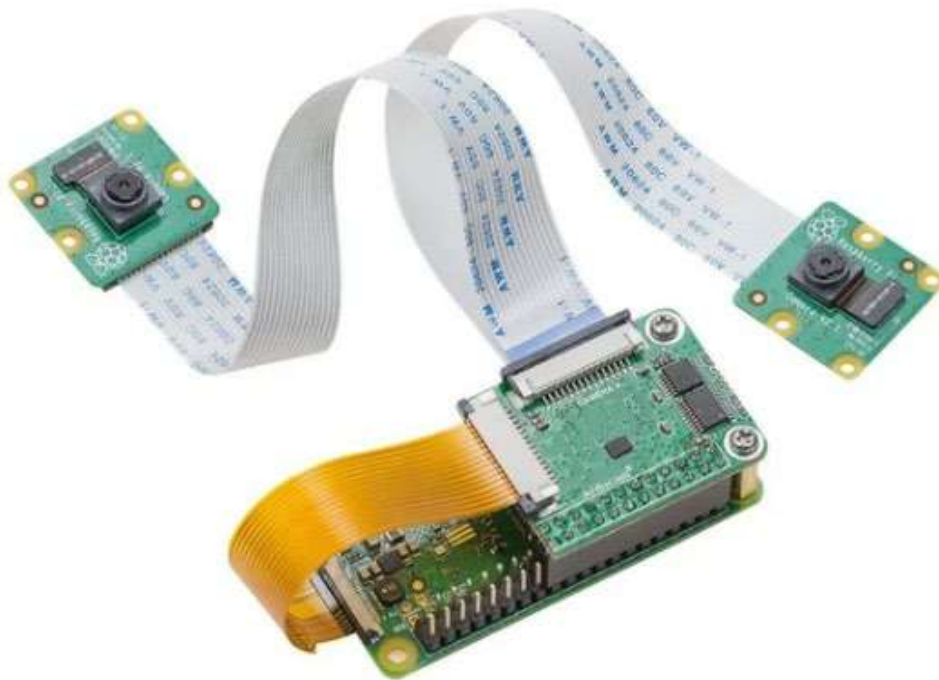
4.3.11 Multi-kamera adapter

Kada je donesena odluka koju kameru želimo koristiti, potrebno je dvije odabrane kamere povezati s Raspberry Pi računalom iz razloga što su potrebne minimalno dvije kamere kao bi pretražili sve horizontalne zidove pored kojih robot prolazi. Budući da je za povezivanje ove

dvije kamere potrebna dva CSI porta, podsjetimo se da Raspberry Pi Model 4B posjeduje samo je CSI port, potrebno je koristiti određeni adapter koji omogućuje povezivanje većeg broja kamera koji koriste CSI port, na Raspberry Pi računalo. Za ovaj projekt ćemo koristiti Multi Camera Doubleplexer Stereo Module V2, koji omogućava povezivanje dvije kamere koje koriste CSI sučelje, a primarno je namijenjen za Raspberry Pi platformu. Ovaj adapter radi na način da je jedna kamera uključena vrlo kratko vrijeme nakon čega se isključuje prava a uključuje druga kamera te se na taj način stvara privid da obje kamere rade u isto vrijeme, što svakako nije slučaj. Uz navedeni multi-kamera adapter, postoje izvedbe koji omogućavaju spajanje i više od dvije kamere koje koriste CSI sučelje za komunikaciju, a prikaz multi-kamera adaptera kojeg koristimo u ovom projektu dan je na sljedećoj slici.



Slika 4.28 Prikaz Camera Doubleplexer Stereo Modula V2 [21]



Slika 4.29 Prikaz povezivanja Camera Doubleplexer Stereo Modula V2 s perifernim uređajima [21]

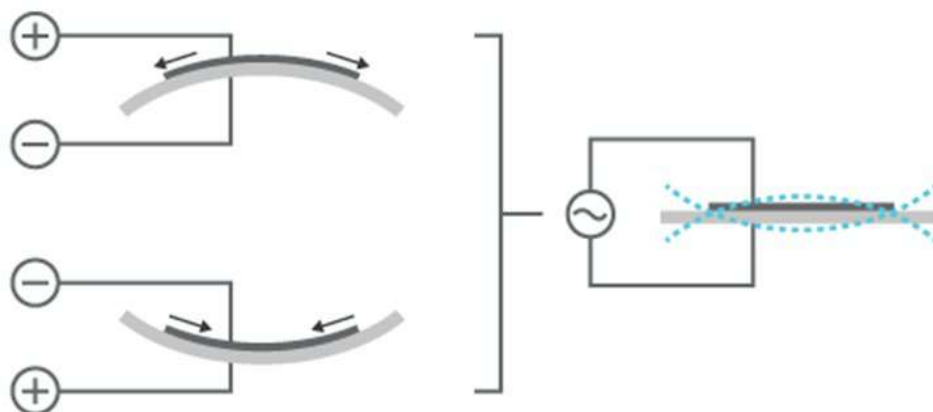
4.3.12 Zvučnik

Kada robot prepozna određenu metu, potrebno je da emitira zvučni i svjetlosni signal u trajanju od 5 sekundi. Zvučni signal realiziramo pomoću arduino zvučnog senzora (eng. buzzer) koji je prikazan na sljedećoj slici.



Slika 4.30 Prikaz zvučnika korištenog u projektu [22]

Iz prethodne slike je vidljivo da se ovaj uređaj sastoji samo od dva pina, čije je funkcija napajanje naponima u intervalu od 4 V do 8 V. Duža nožica uređaja označava pin na koji se dovodi veći naponski potencijal. Pojednostavljena unutarnja građa ovog uređaja je prikazana na sljedećoj slici.

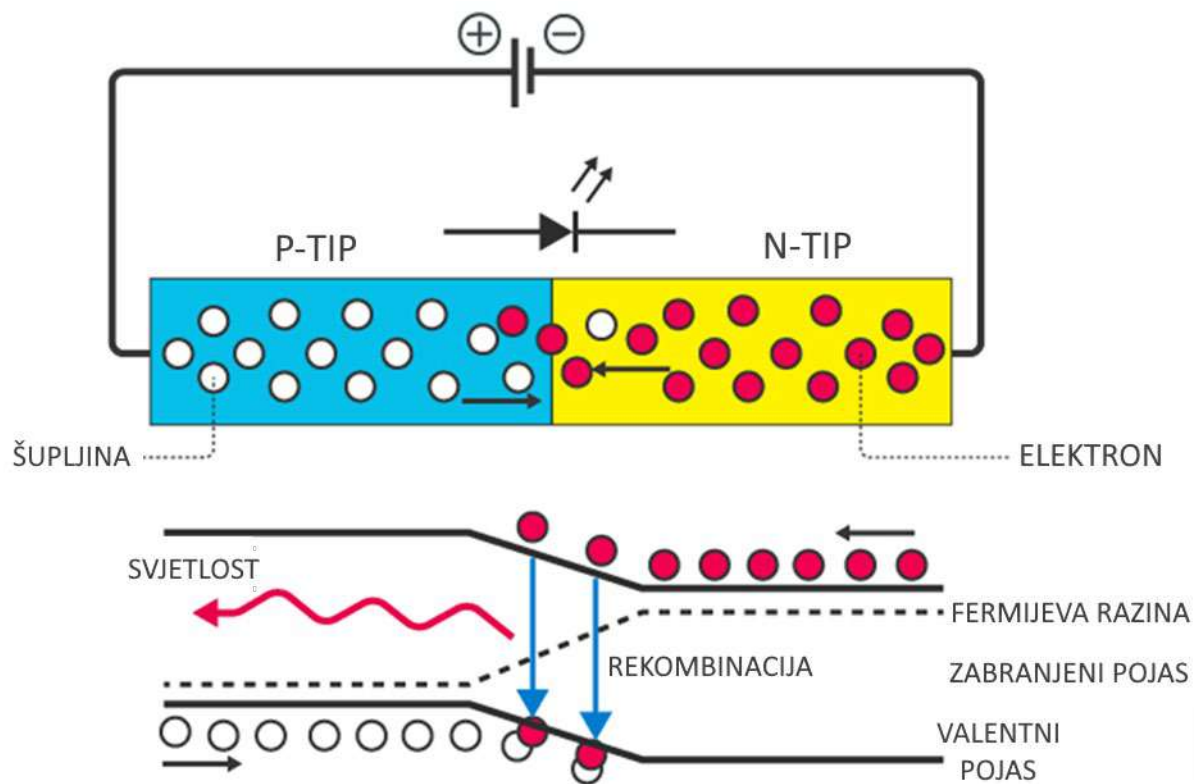


Slika 4.31 Prikaz širenja i skupljanja piezoelktričnog elementa ovisno o naponu [23]

Iz slike je vidljivo da pinovi napajanja zatvaraju strujni krug preko piezokeramičkog materijala čije je svojstvo sferno širenje i skupljanje u određenom smjeru ovisno o polaritetu napona koji je na njega doveden, te metala koji je povezan s piezokeramičkim materijalom koji se zakrivljuje zahvaljujući povezanosti s piezokeramičkim materijalom te na taj način emitira zvuk određene frekvencije.

4.3.13 Ledica

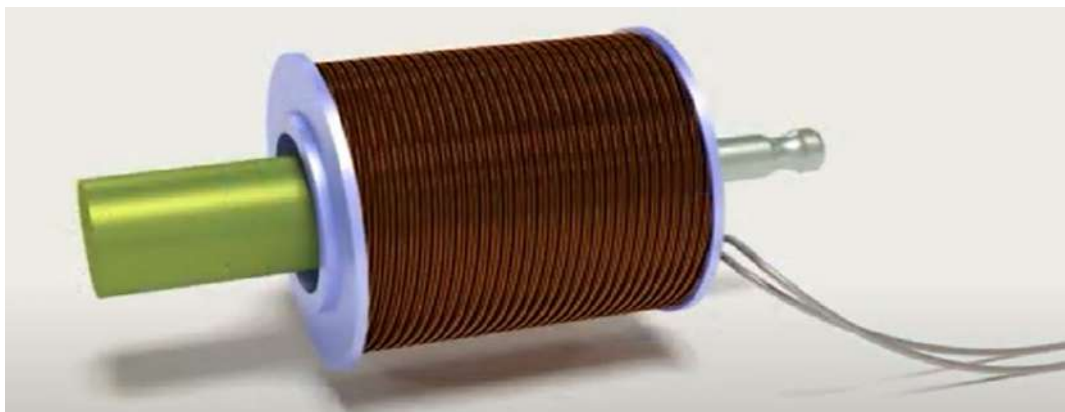
Svjetlosni signal kojeg robot treba emitirati određeni vremenski period realiziran je pomoću svijetleće diode, ledice (eng. LED). Ledica se sastoji od dvije nožice koje služe za dovođenje napajanja, pri čemu se na duži izvod pina dovodi veći električni potencijal napona u intervalu od 2 V do 3,3 V. LED dioda se sastoji od dva tipa poluvodiča p-tipa kojem su većinski nosioci naboja šupljine nabijene pozitivnim nabojem i n-tipa poluvodiča, kojem su većinski nosioci naboja elektroni nabijeni negativnim nabojem. Kada dovedemo određenu razinu „pozitivnog“ naboja na poluvodič p-tip-a, čiji je izvod duži pin napajanja ledice, te određenu razinu „negativnog“ naboja na poluvodič n-tip-a, čiji je izvod kraći pin napajanja ledice, dolazi do rekombinacije između elektrona i šupljina, odnosno prelaska elektrona iz višeg energetskog stanja u niže energetsko stanje pri čemu se emitira foton svijetlosti. Emitirana svijetlost nije uvijek u vidljivom spektru ljudskog oka, npr. može biti u infracrvenom ili ultraljubičastom spektru koji nije vidljiv ljudskom oku, stoga je uz određene napone potrebno odabrati i određene tipove poluvodiča kako bi emitirali željenu svijetlost.



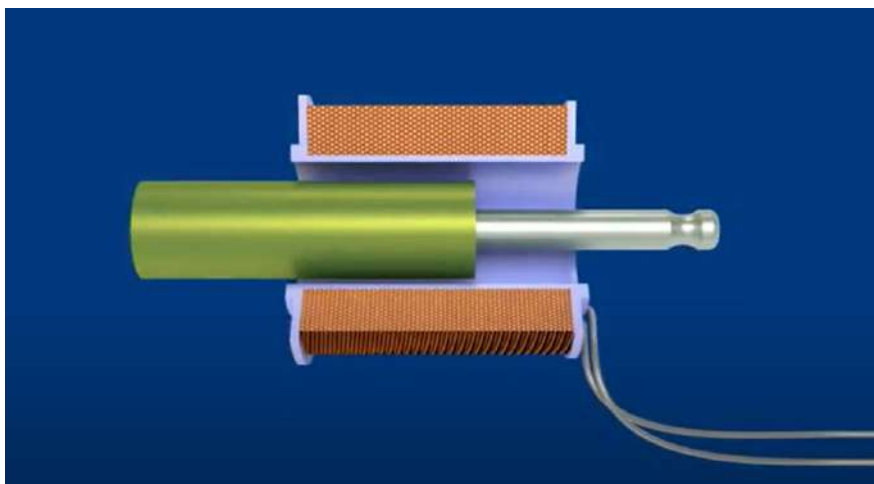
Slika 4.32 Prikaz principa emitiranja svjetlosti ledice [24]

4.3.14 Solenoid

Solenoid je komponenta čija je funkcija izgurivanje paketa pomoći iz spremnika robota u kojem se nalaze. Unutarnja građa solenoida prikazana je su nastavku.

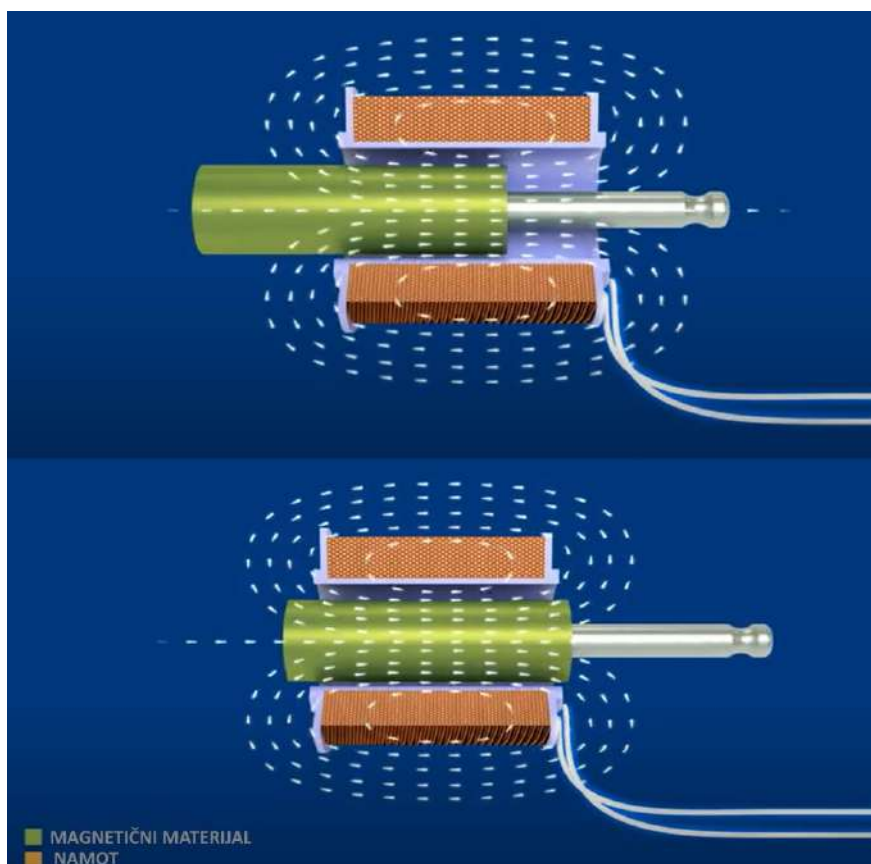


Slika 4.33 Prikaz unutarnje građe solenoida [25]



Slika 4.34 Solenoid bez napajanja u početnom položaju [25]

Iz prethodne slike vidljivo je da se solenoid sastoji od pinova na napajanje, zavojnice, feromagnetizirajućeg materijala, te izlazne osovine. Princip rada je solenoida vrlo je jednostavan. Kada na pinovima napajanja nije prisutno napajanje odgovarajućom snagom električne energije sustav se nalazi u početnom položaju. Kada je na pinove napajanja dovedena električna energija odgovarajuće snage, u zavojnici dolazi do kreiranja elektromagnetskog polja koje magnetizira i privlači feromagnetski materijal te se cjelokupna izlazna osovina giba u desno.



Slika 4.35 Prikaz pomjeranja osovine solenoida u konačan položaj [25]

Kada solenoidu prekinemo dovod napajanja (npr. preko sklopke, tranzistora, u našem projektu koristimo kontroler motora) izlazna osovina se vraća u početni položaj zahvaljujući opruzi koja se montira na način da sila opruge djeluje u istom smjeru i suprotnoj orijentaciji od orijentacije elektromagnetskog polja koje pomjera izlaznu osovinu solenoida u krajnji desni položaj.

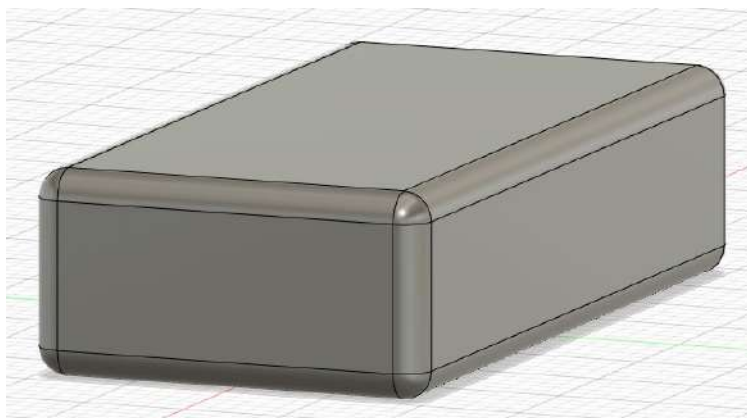
Solenoid koji se koristi u projektu nosi oznaku TAU-0730B, a prikazan je na sljedećoj slici.



Slika 4.36 Prikaz solenoida TAU-0730B [26]

4.3.15 Paket pomoći

Paket pomoći koji je potrebno dostaviti metama, dimenzija je 1cm x 0.5 cm x 2 cm, a prikazan je na narednoj slici.

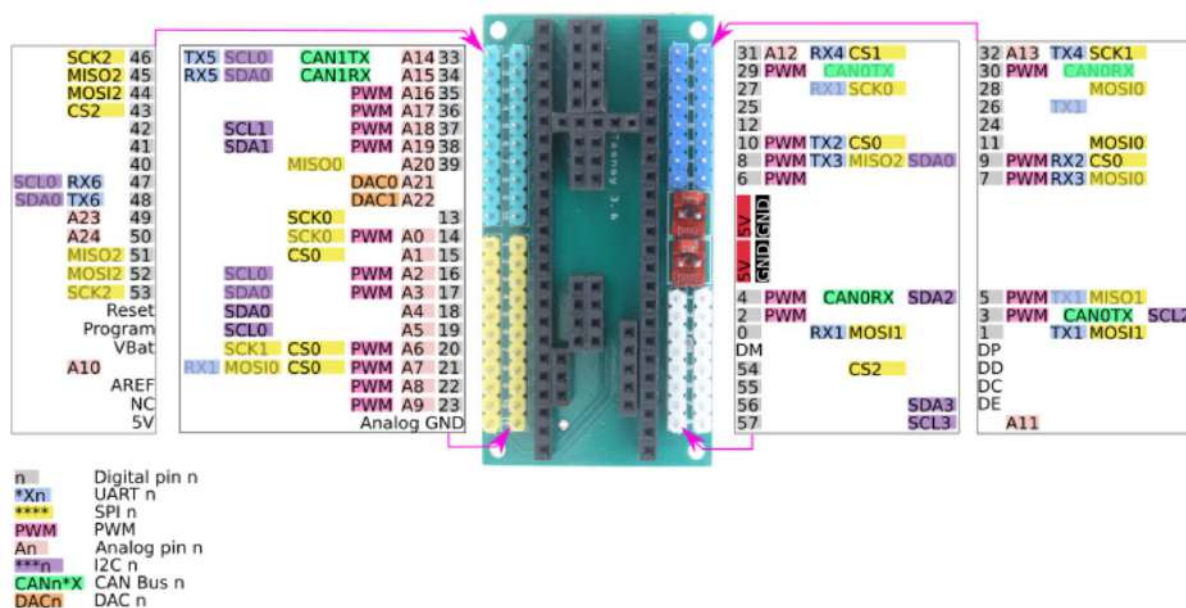


Slika 4.37 Prikaz paketa pomoći

4.3.16 Adapter za Teensy 3.5

Kako bi bio omogućen jednostavniji pristup pinovima mikrokontrolera Teensy 3.5, koristimo adapter na kojeg je samo potrebno povezati Teensy 3.5. Adapter ne predstavlja ništa više od

običnog razvoda žica koje su raspoređene na određeni način. Prikaz adaptera s funkcijama pinova koje su proslijeđene s mikrokontrolera Teensy 3.5 dan je u nastavku.



Slika 4.38 Prikaz adaptera za Teensy 3.5 s funkcionalnostima pinova [11]

4.3.17 Raspored komponenti

Cilj rasporeda komponenti je postići što je veću moguću stabilnost robota, kao i jednostavnost pristupa određenim komponentama, stoga je cilj nateže komponente (baterija, motori) što je više moguće približiti točki centra mase robota koja je poželjno da se nalazi na najnižoj mogućoj razini na pravcu rotacije robota u horizontalnoj ravnini (X-os).

Žiroskop se mora nalaziti u središtu rotacije robota kako bi kut rotacije žiroskopa odgovarao stvarnoj rotaciji robota te na mjestu gdje nije značajnije prisutno magnetno polje koje može stvarati smetnje u radu žiroskopa.

Za detektiranje meta potreban je po jedan senzor za određeni tip žrtve, s lijeve i desne strane robota iz razloga što lijeva i desna strana tijekom manevriranja robota po labirintu imaju uvid u sve zidove labirinta pored kojih je robot prošao, što znači da ćemo i s lijeve i s desne strane robota imati po jedan par senzora koji čini termalna i optička kamera čiji se senzor treba nalaziti na 7 cm od horizontalne podloge na kojoj se robot nalazi kako bi imali najbolji uvid u položaj žrtvi.

Sa sve četiri bočne strane robota imat ćemo po dva senzora udaljenosti (ukupno osam) kako bi robot mogao određivati svoj smjer u odnosu na zidove labirinta, što proizlazi iz činjenice da je smjer nekog pravca određen sa dvije točke koje će u našem slučaju mjeriti senzori udaljenosti.

4.4 Izrada tijela robota

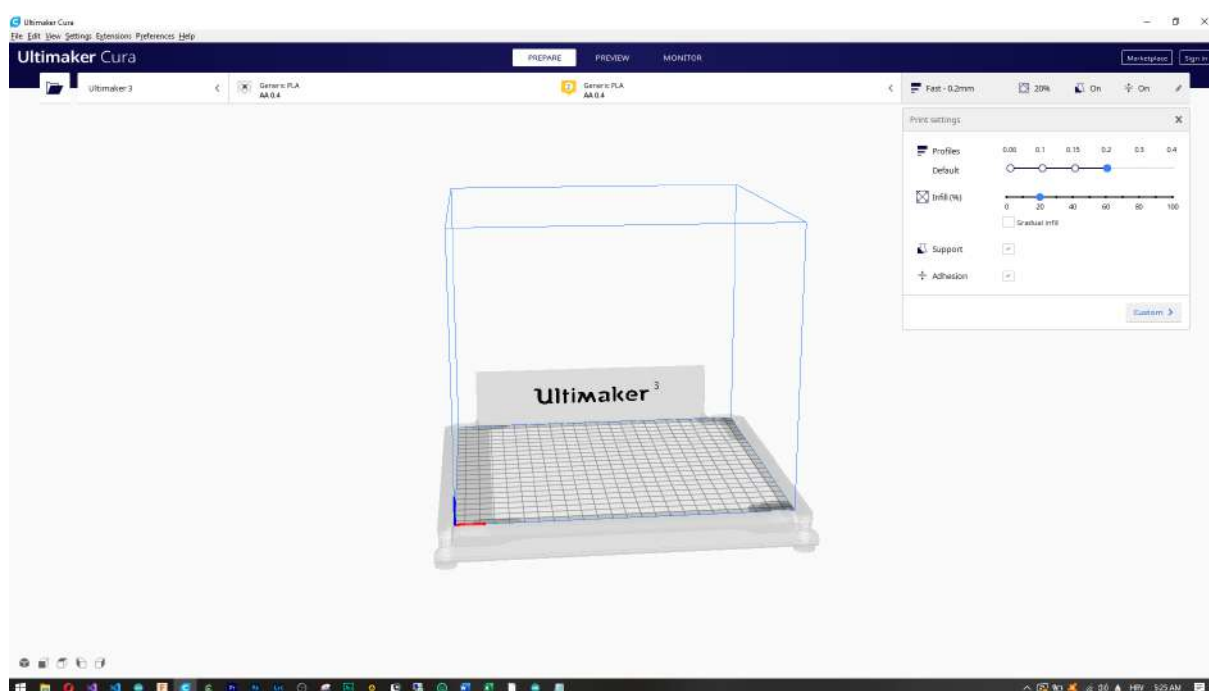
4.4.1 Dizajniranje i priprema 3D modela za postupak 3D printanja

Tijelo robota dizajnirano je u programu kompanije Autodesk koji se naziva Fusion 360, čije iznimno učinkovito sučelje omogućuje vrlo jednostavno i brzo kreiranje najsloženijih 3D objekata. Ovaj program uz 3D dizajniranje odnosno CAM (eng. Computer-Aided Design) omogućuje i kreiranje g-koda i simulaciju cjelokupnog postupka obrade materijala pomoću CNC (eng. Computerized Numerical Control) uređaja. Objašnjenje rada u Fusion 360 programu nije tema ovog diplomskog rada, stoga sve znanje potrebno za rad u ovom programu vrlo je dostupno na globalnoj internet mreži. Besplatnu probnu verziju kao i edukacijsku te sve ostale verzije ovog programa možete preuzeti na službenim stranicama kompanije Autodesk.

Nakon što je tijelo robota dizajnirano prema zahtjevima navedenim u poglavlju „2. rescue maze -analiza pravila i okruženja“ u programu Fusion 360, konstruirani 3D model potrebno je izvesti kao .STL datoteku na osnovu koje ćemo generirati g-kod za 3D printer.

G-kod generiramo pomoću programa Ultimaker Cura koji je besplatan za korištenje.

Početni prozor programa Ultimaker Cura prikazan je na sljedećoj slici.



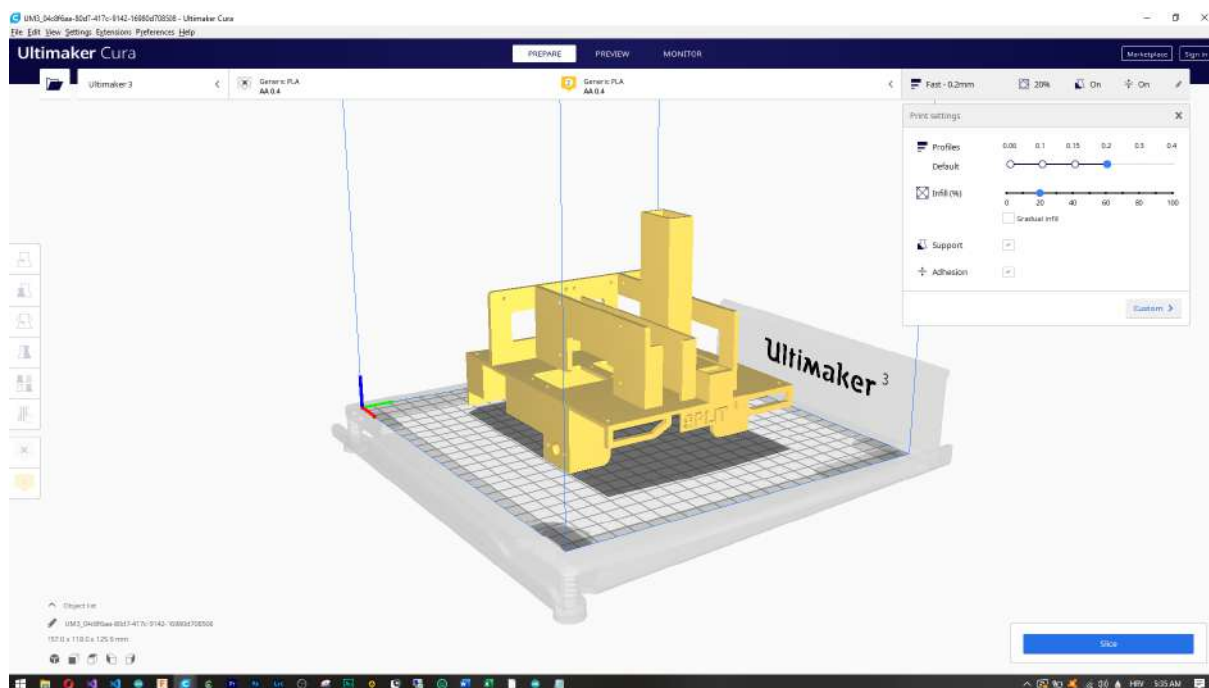
Slika 4.39 Prikaz početnog sučelja programa Ultimaker Cura

U ovom prozoru potrebno je odabrati na kojem 3D printeru će se vršiti postupak 3D printanja kao i .STL model koji želimo 3D printati.

Za ovaj projekt koristimo Ultimaker 3, 3D printer.

Objašnjavanje postavki vršit će se na 3D modelu donje konstrukcije robota.

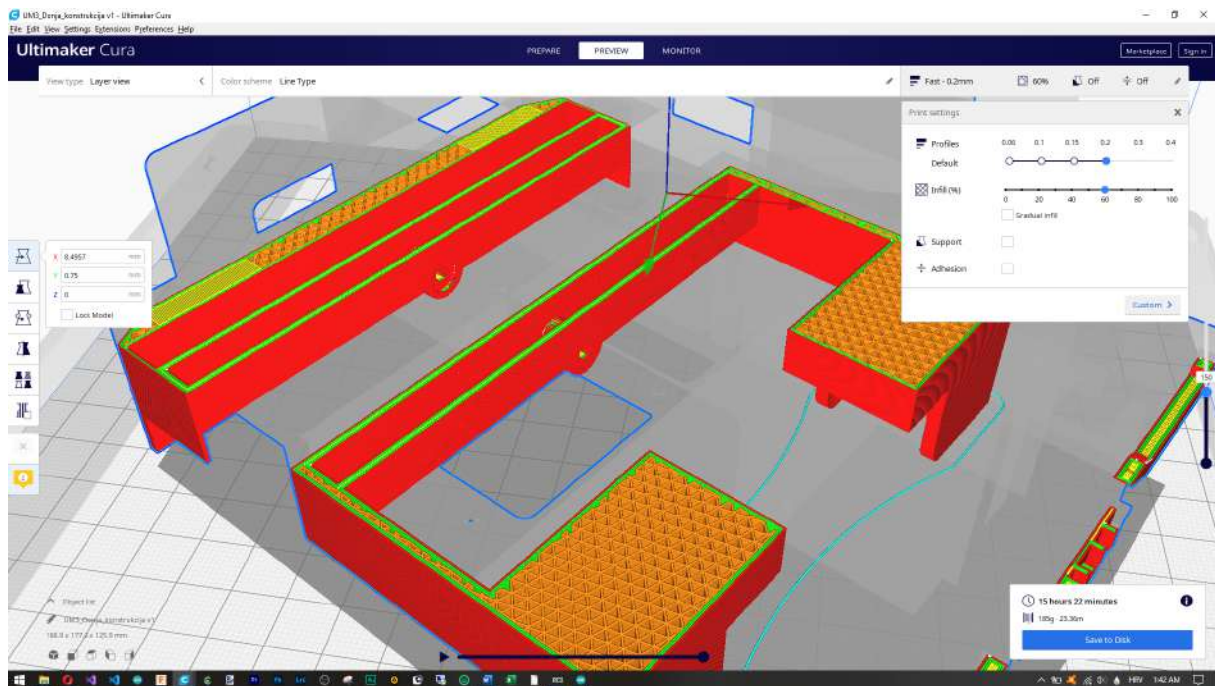
Nakon odabranog modela potrebno je odabrati postavke postupka 3D printanja, pri čemu se dotičemo osnovnih postavki koje su prikazane u gornjem desnom kutu naredne slike, a nose nazive „Profiles“, „Infil“, „Support“ i „Adhesion“.



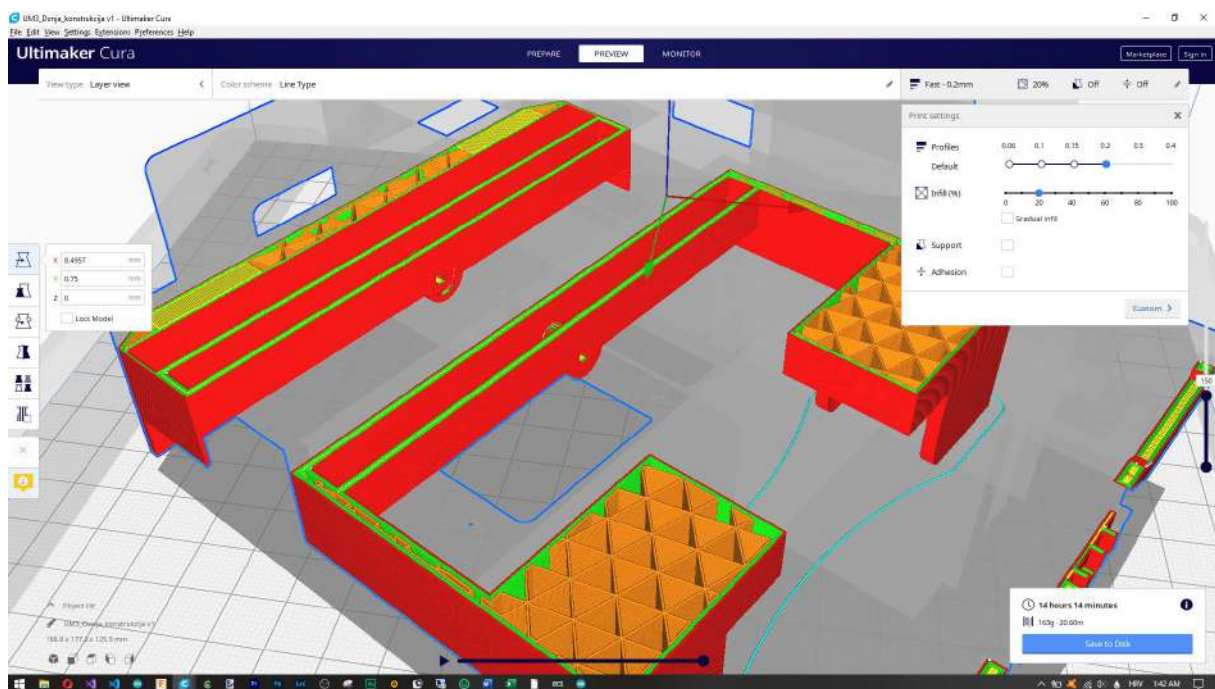
Slika 4.40 Prikaz 3D modela učitano u sučelje programa Ultimaker Cura

Postavka „Profiles“ označuje promjer niti printnog materijala 3D printera odnosno ukoliko želimo precizniji objekt odabrat ćemo manji promjer printne niti, što povećava vrijeme printanja objekta.

Postavka „Infil“ označava postotak ispunjenosti zatvorenog volumena određenog objekta. Ovom postavkom utječemo na čvrstoću odabranog 3D modela.



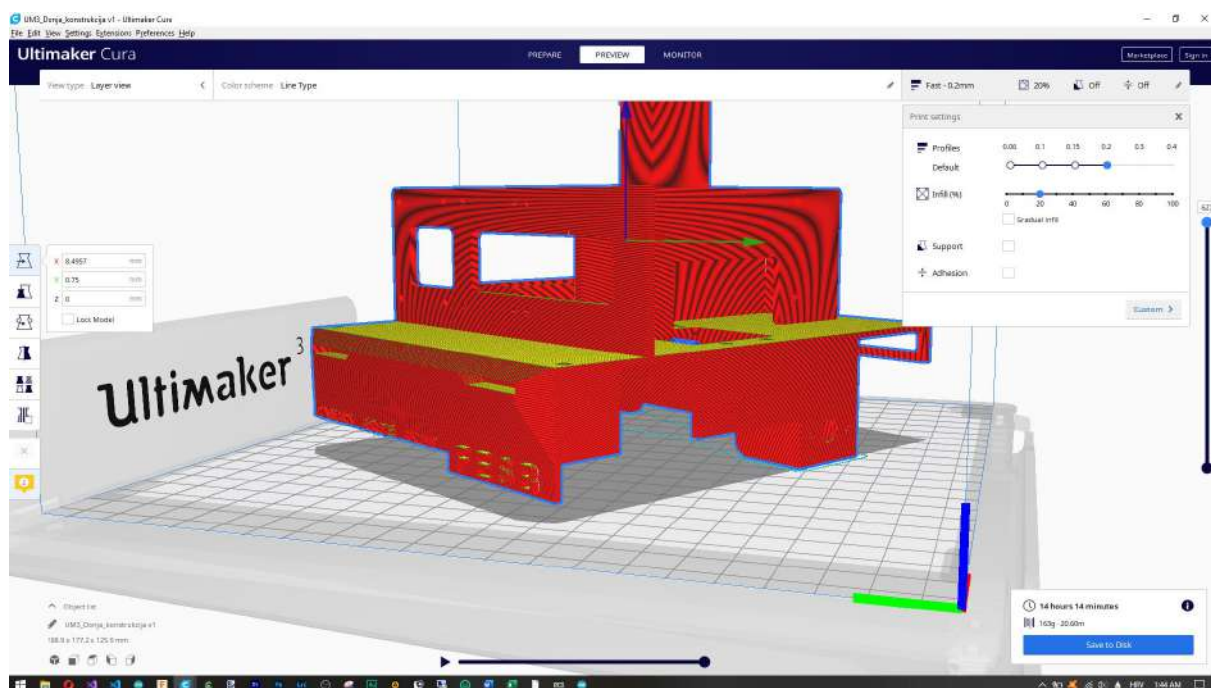
Slika 4.41 Prikaz 3D modela kada je postavka „Infil“ na 60% vrijednosti



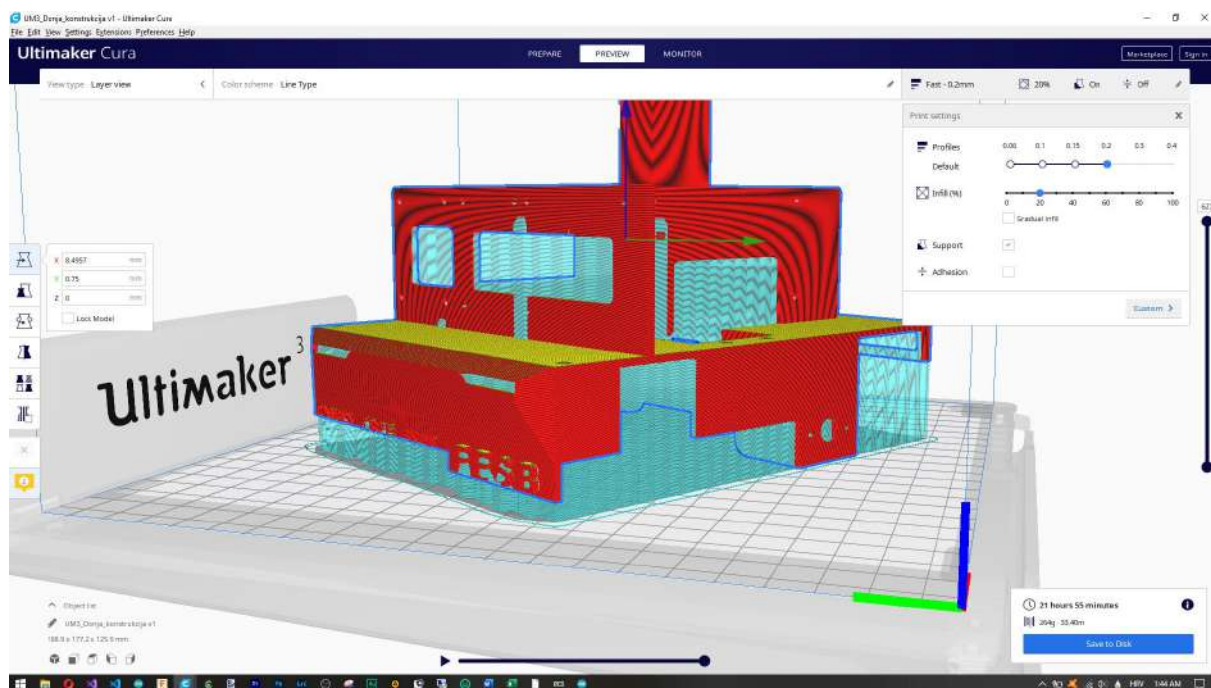
Slika 4.42 Prikaz 3D modela kada je postavka „Infil“ na 20% vrijednosti

Postavka „Support“ treba biti označena ukoliko na odabranom 3D modelu postoje dijelovi koji „lebde u zraku“ iz razloga ako određeni dijelovi 3D modela lebde, nit kojom 3D printer vrši printanje nema se na što osloniti, stoga postoji velika vjerojatnost da odabrani model neće biti isprintan na željeni način. Opcija „Support“ gradi potporu za dijelove koji lebde kao bi se nit 3D printera imala na što osloniti. Kada je 3D objekt isprintan, ove potpore se

povlačenjem u raznim smjerovima vrlo jednostavno mehanički odstranjuju s isprintanog objekta.

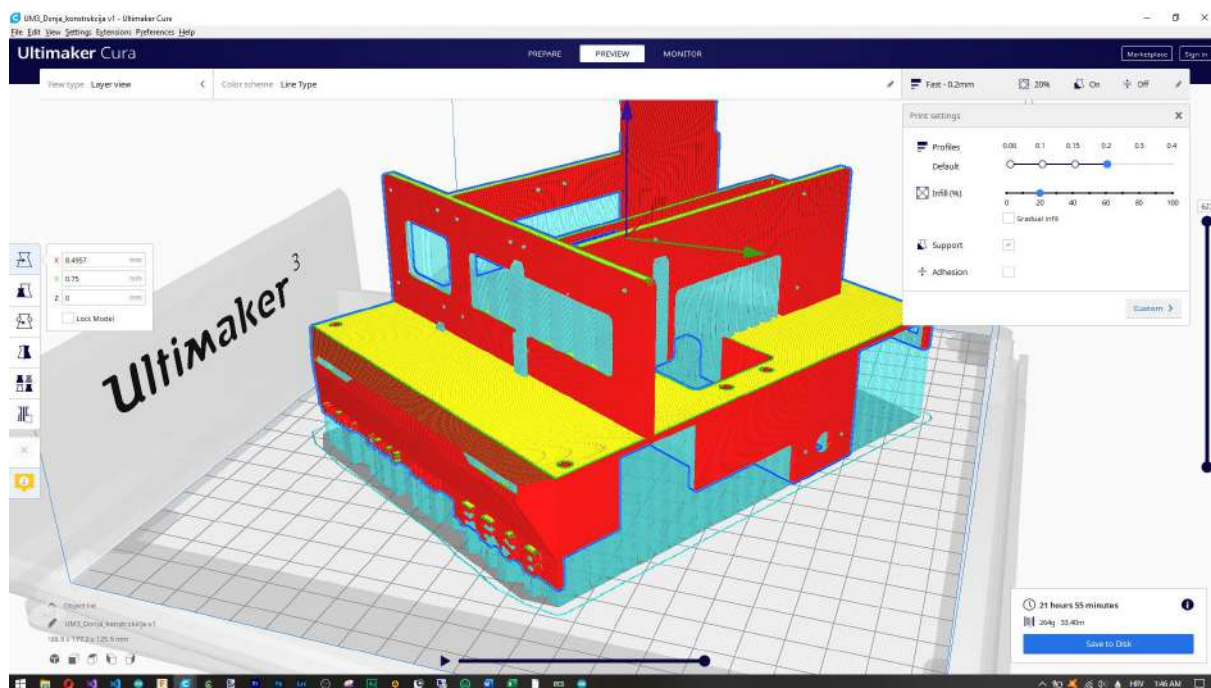


Slika 4.43 Prikaz 3D modela kada je postavka „Support“ isključena

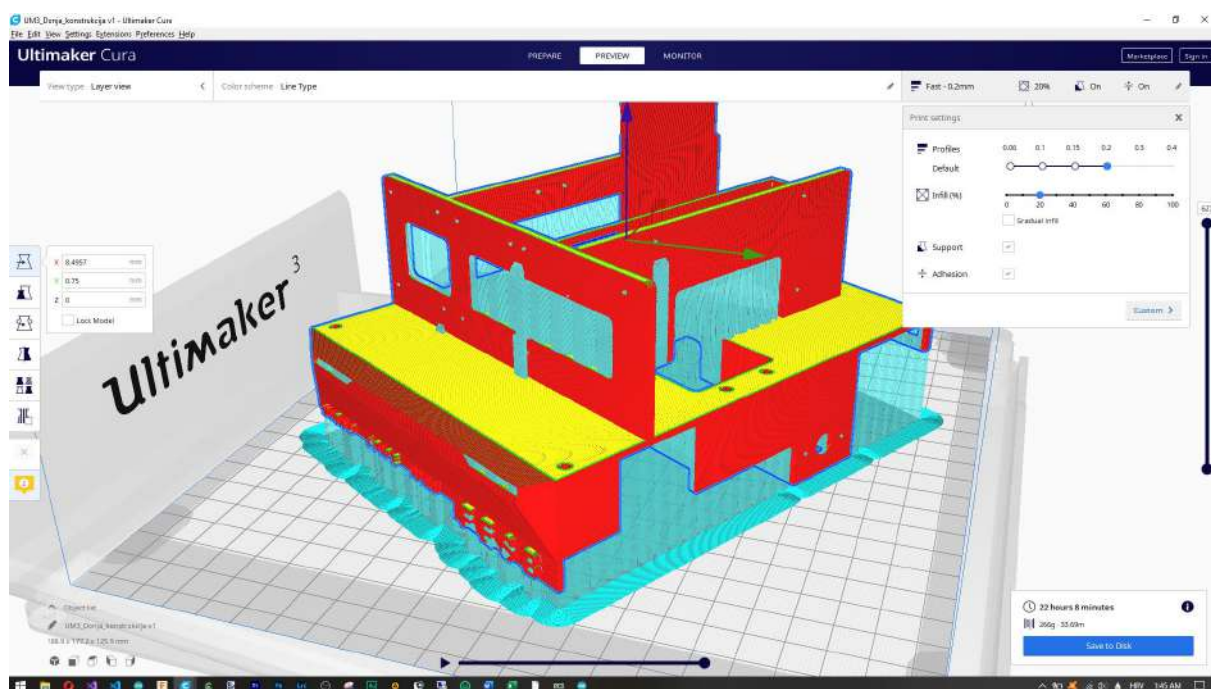


Slika 4.44 Prikaz 3D modela kada je postavka „Support“ uključena

Postavka „Adhesion“ dodatno pričvršćuje koture prvog sloja printanja odabranog 3D objekta za podlogu na kojoj se vrši printanje. Poželjno ju je koristiti pri svakom printanju, ali ne i neophodno.



Slika 4.45 Prikaz 3D modela kada je postavka „Adhesion“ isključena



Slika 4.46 Prikaz 3D modela kada je postavka „Adhesion“ uključena

Nakon što su postavke odabrane na željni način potrebno je pritisnuti komandu „Slice“ koja se nalazi u donjem desnom kutu prethodne slike kako bi se 3D model preveo u strojni jezik razumljiv odabranom 3D printeru.

Cjelokupni postupak printanja određenog modela moguće je simulirati u „Prewiew“ prozoru Ultimaker Cura programa.

4.4.2 Postupak 3D printanja

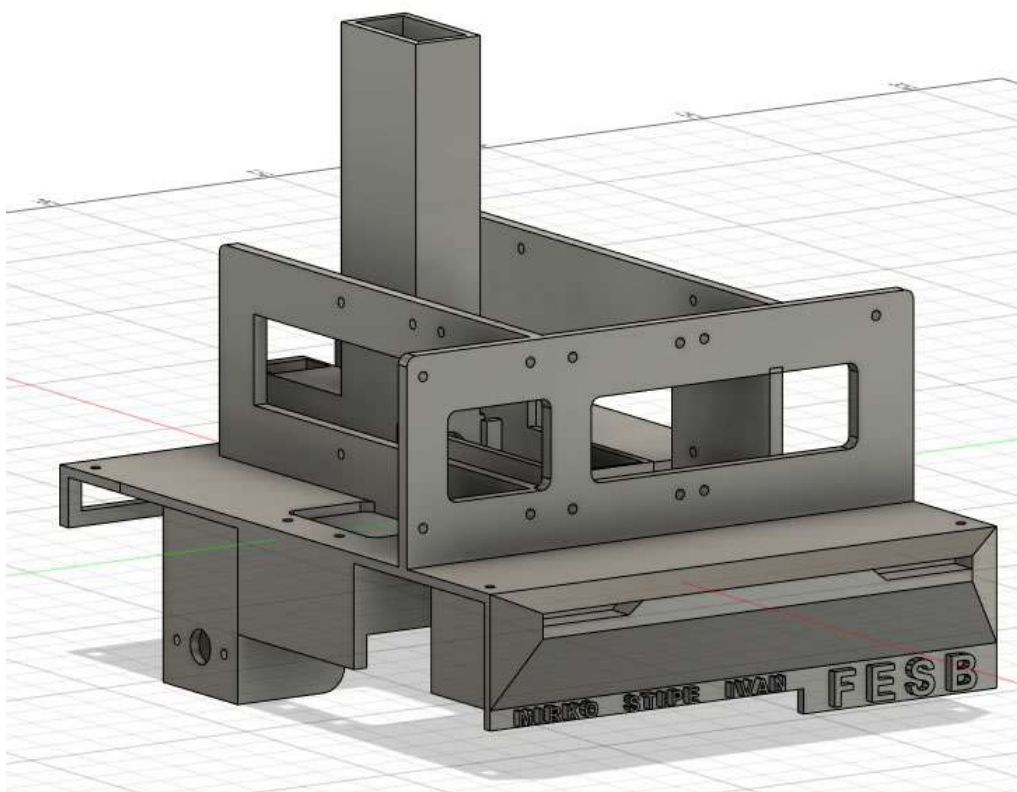
Nakon što je g-kod generiran od strane Ultimaker Cura programa, potrebno je da ga 3D printer očita. Ovisno o 3D printeru g-kod datoteku možemo pomoću 3D printera očitavati s USB diska, memorijske kartice ili pak preko Cloud strukture. Za ovaj projekt očitavanje vršimo preko USB diska koji za prvu uporabu mora biti formatiran protokolom FAT 32. Nakon što je USB disk formatiran na odgovarajući način, na njega prebacujemo g-kod 3D objekta, te USB disk uključujemo u USB utor koji se nalazi na 3D printeru. Prije samog postupka printanja potrebno je odabrati adekvatan materijal od kojeg će se izraditi tijelo robota. Za naš projekt odabran je PLA materijal (eng. Polylactic Acid), odnosno filament, koji na odgovarajući način postavljamo na 3D printer te pokrećemo postupak 3D printanja.

Poželjno je imati uvid u cjelokupan postupak 3D printanja određene komponente, koje možemo ostvariti npr. pomoću Raspberry Pi računala i kompatibilne kamere, ali i preko kamere koje već dođu ugrađene u sam 3D printer što je karakteristika skupljih izvedbi određenih modela 3D printera. Podloga na kojoj se printa mora biti ravna i bez ikakvih ostataka prethodnih printanja ili čestica prašine, kako bismo znatno unaprijedili izgled da se željni objekt isprinta na željeni način. Svakako je najvažnije pratiti prvih nekoliko slojeva objekta koji se printa jer prema stečenom iskustvu, najviše „grešaka“ 3D printera se događa na samom početku printanja, okvirno do petog horizontalnog sloja kojeg 3D printer napravi.

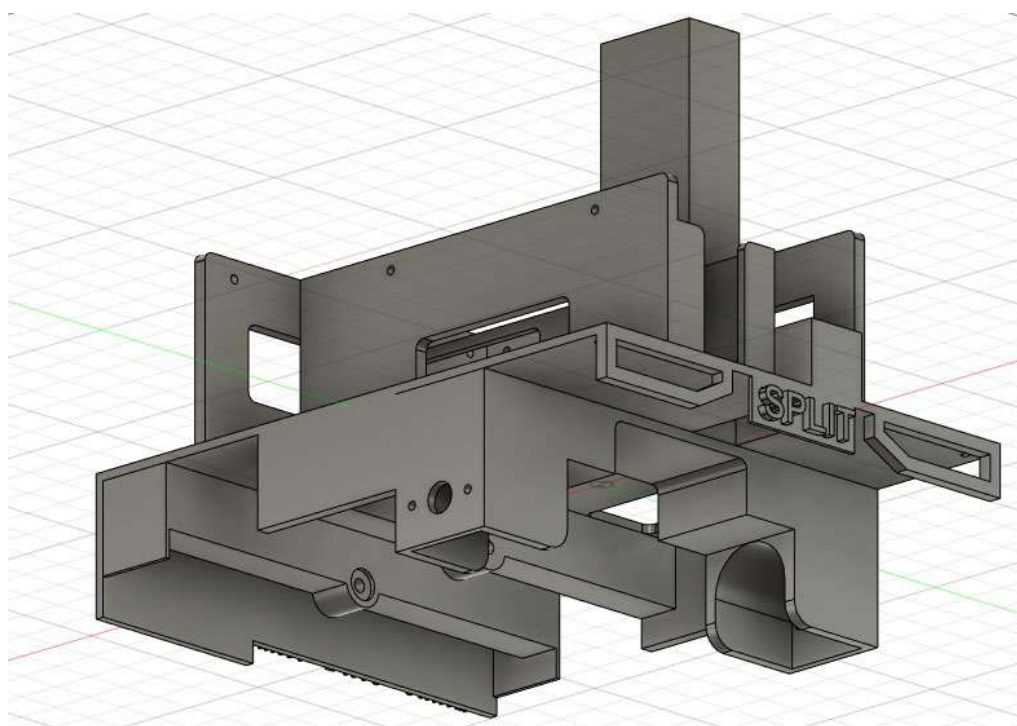
Nakon što je 3D model uspješno isprintan potrebno ga je ukloniti s podloge 3D printera.

4.4.3 Komponente robota koje čine tijelo robota

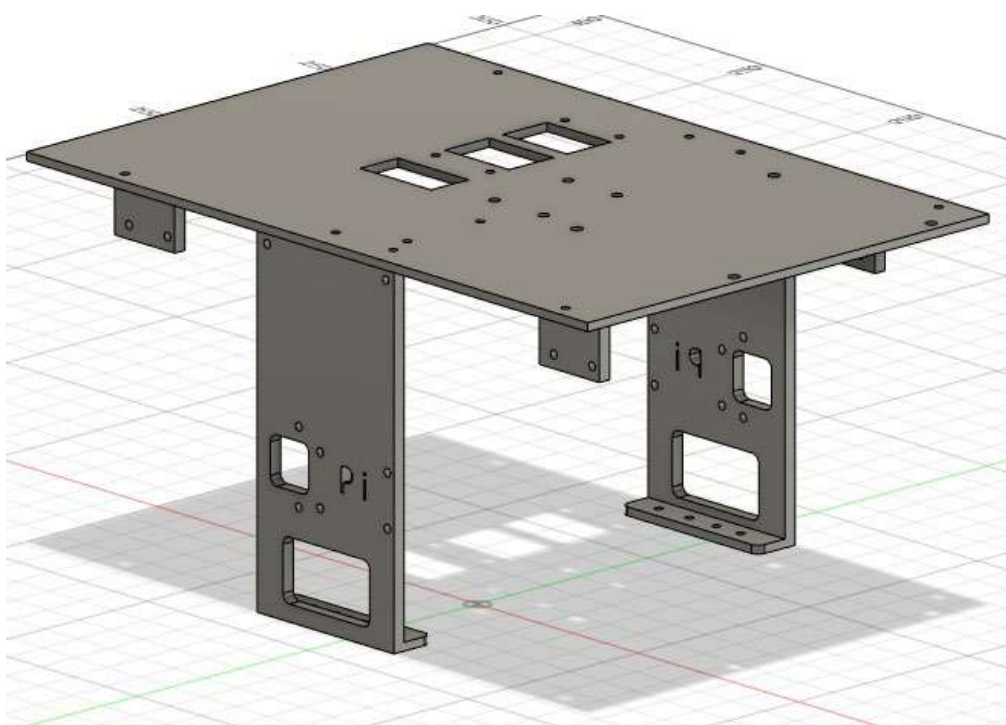
Dvije su temeljne komponente od kojih se sastoji tijelo robota, a to su donja i gornja konstrukcija, a prikazane su u nastavku.



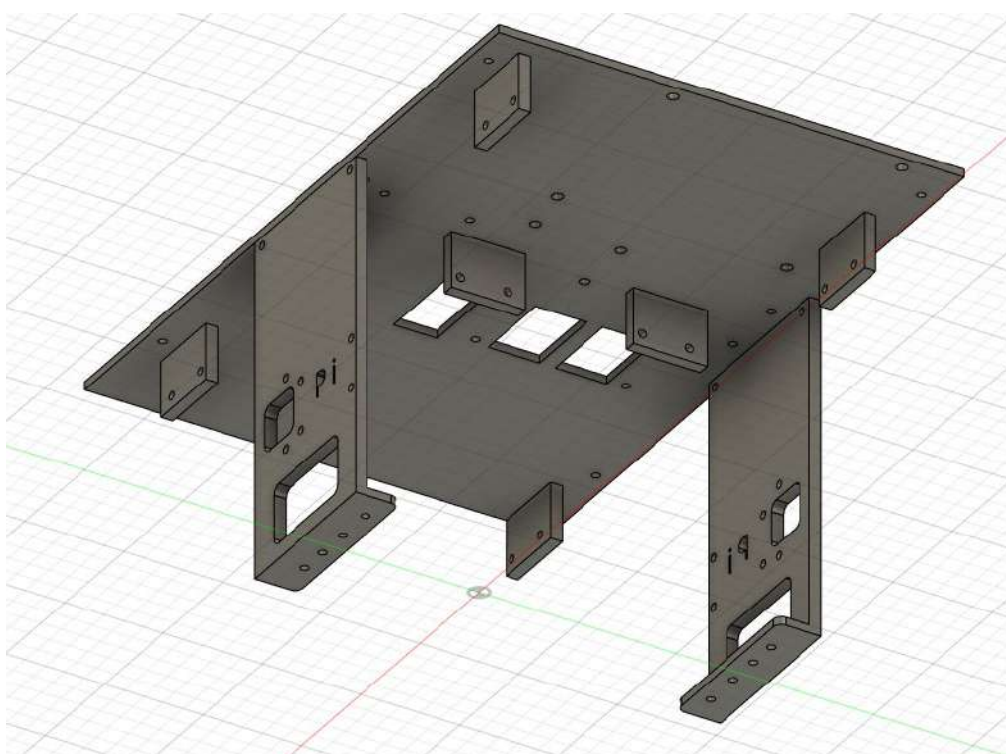
Slika 4.47 3D model donje konstrukcije robota



Slika 4.48 3D model donje konstrukcije robota



Slika 4.49 3D model gornje konstrukcije robota

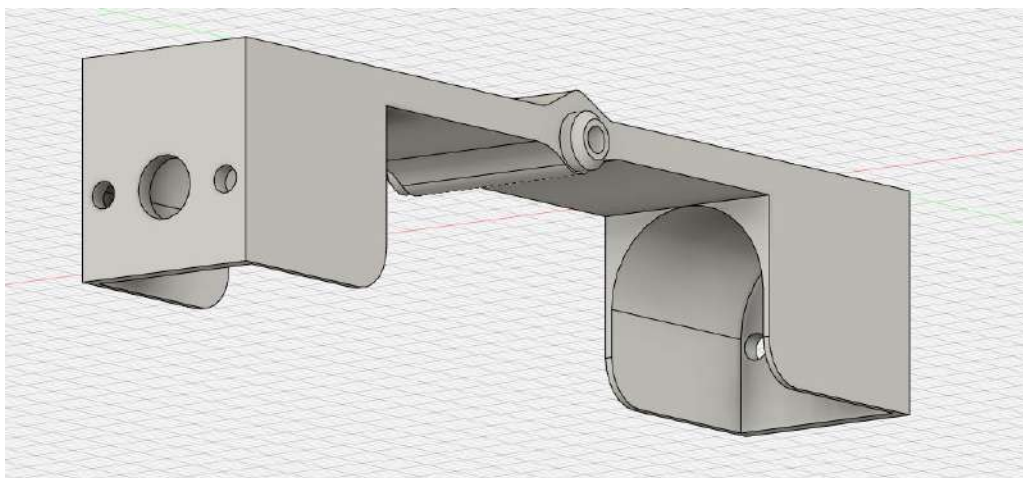


Slika 4.50 3D model gornje konstrukcije robota

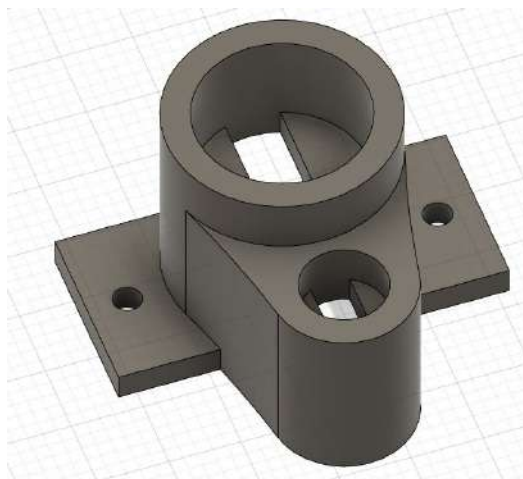
Uz gornju i donju konstrukciju tijelo robota čine amortizirajuća osovina, četiri kotača te držač za zvučnik i ledicu koji su prikazani u nastavku.



Slika 4.51 3D model kotača robota

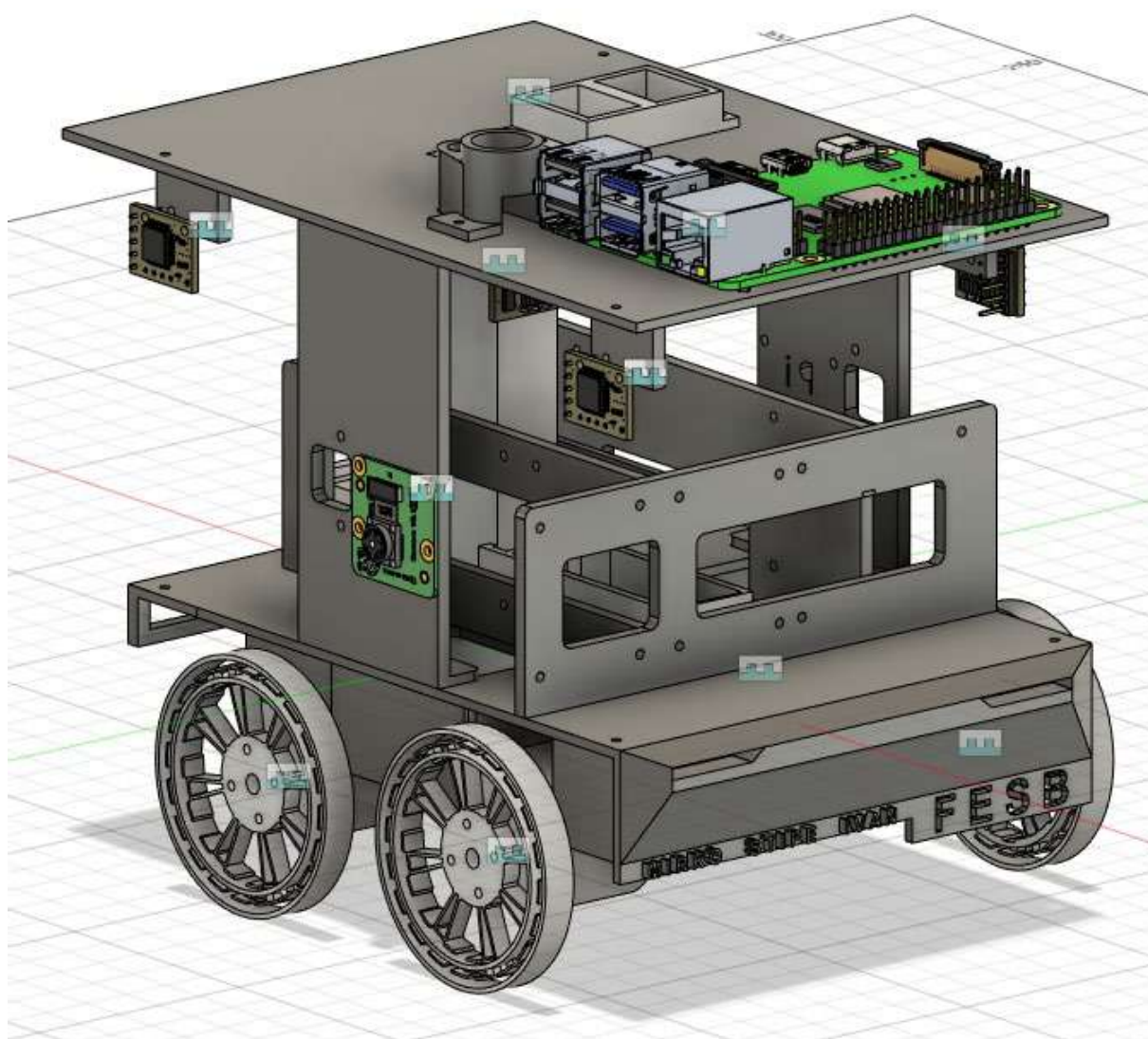


Slika 4.52 3D model prednje amortizirajuće osovine robota

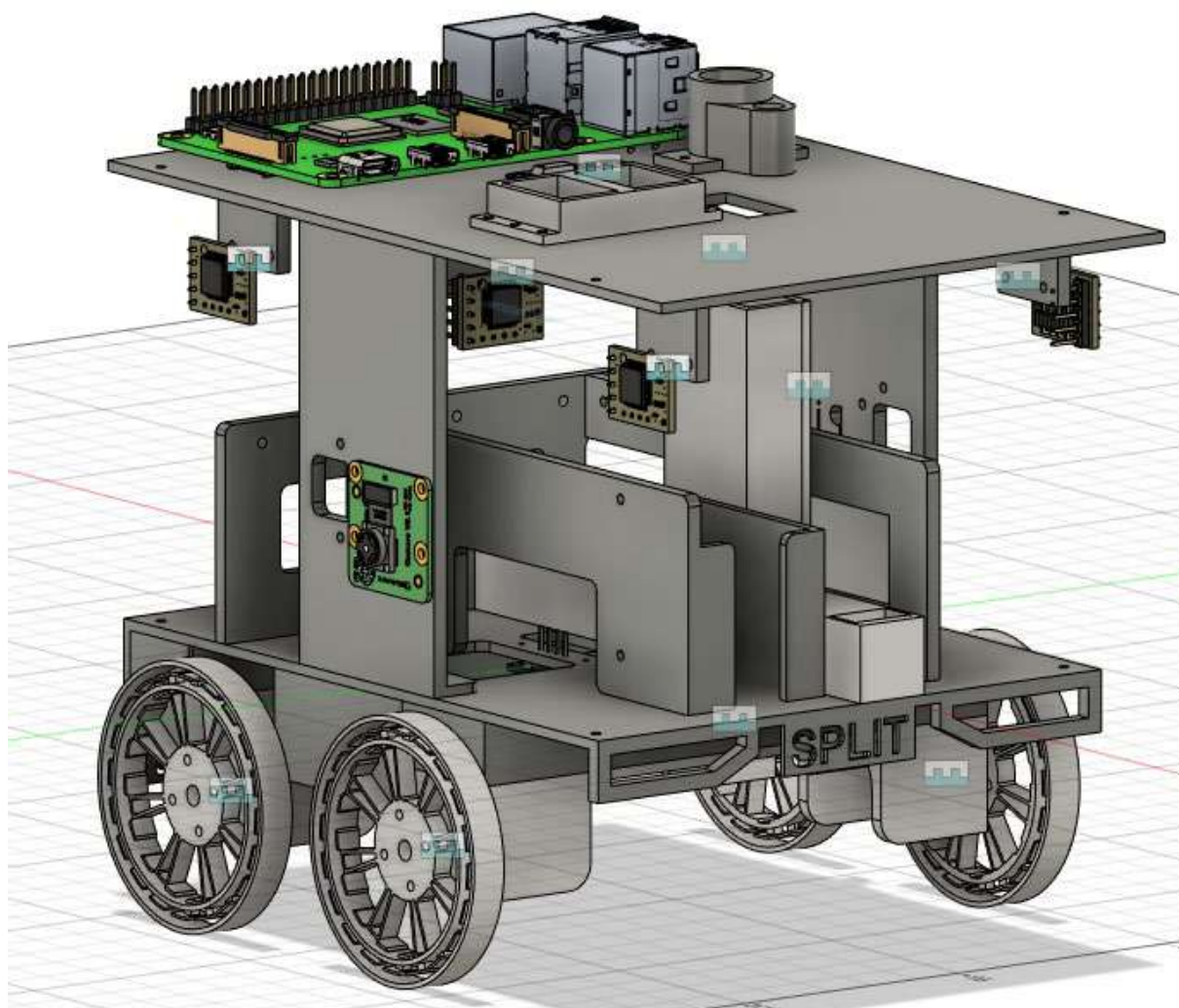


Slika 4.53 3D model držača za LED i zvučnik

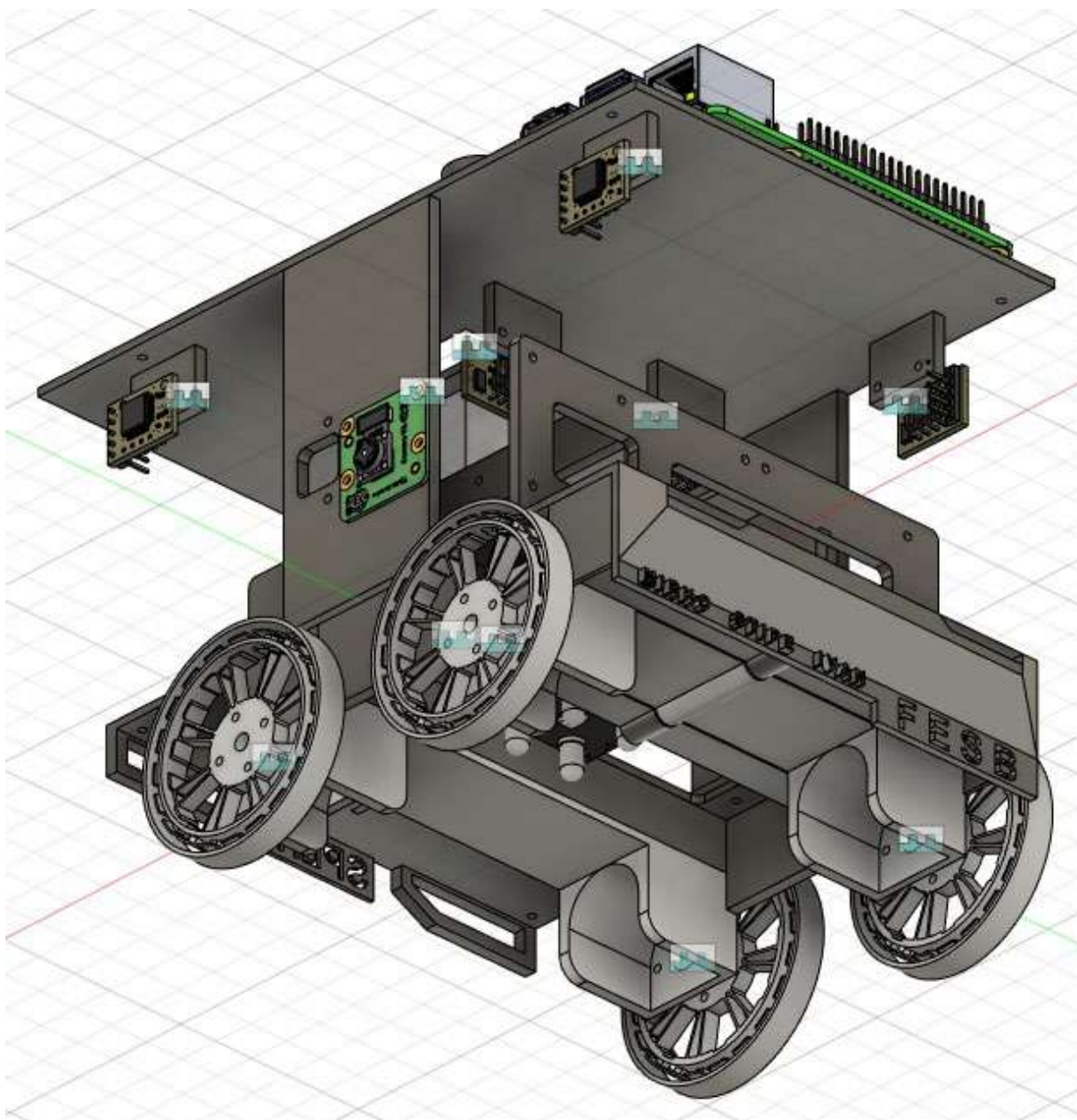
Prikaz 3D modela tijela robota s određenim brojem hardverskih komponenti robota dan je u nastavku.



Slika 4.54 3D model tijela robota s određenim brojem hardverskih komponenti



Slika 4.55 3D model tijela robota s određenim brojem hardverskih komponenti



Slika 4.54 3D model tijela robota s određenim brojem hardverskih komponenti, pogled s donje strane robota

3D modeli komponenti korištenih u ovom projektu dostupni su na referenci [31]

4.5 Električna shema robota

Električna shema robota ljudskom oku nije dovoljno pregledna iz razloga što se nalazi jako puno komponenti i žičanih kontakata na jako malom prostoru te nakon printanja diplomskog rada shema bi postala beskorisna. Objašnjenja svih korištenih pinova i načina spajanja pojedinih komponenti dana su u ovom poglavlju. Također imenovanje pinova komponenti prisutno i u

programskom kodu što predstavlja dovoljnu količinu informacija za pravilno električno spajanje komponenti robota.

4.6 Cjelokupna programska logika robota

Na robotu se nalaze mikrokontroler i računalo koji komuniciraju preko serijske veze, a povezani su kabelom USB tip A - USB tip micro B.

Na Raspberry Pi računalu se izvršava algoritam za prepoznavanje boja i slova, dok se na Teensy 3.5 vrši očitavanje senzora udaljenosti, boje, orijentacije, temperature. Raspberry Pi i Teensy komuniciraju međusobno slanjem definiranih stringova koji uvjetuju izvršavanje određenih funkcija. Konkretno, ako je Raspberry Pi prepoznao boju ili određeno slovo, dolazi do slanja stringa na Teensy 3.5 putem serijske komunikacije koja je sama po sebi „interrupt“ odnosno radnja koja ima prednost nad svim trenutnim radnjama koje se odvijaju na mikrokontroleru, tako da sve ostale naredbe prestaju sa izvršavanjem dok se ne izvrši funkcija iz interrupta. Primjerice, za prepoznano slovo H koje označava ugroženu žrtvu, zaustavljaju se svi DC motore na 5 sekundi izbacuju dva paketa pomoći, nakon čega se nastavljaju izvoditi naredbe koje su se odvijale prije interrupta. Ista stvar se događa kada Teensy prepozna definiranu specifičnost, dolazi do slanja određenog kodiranog stringa, čije dekodiranje rezultira izvršavanjem željene funkcije, nakon čega se nastavljaju izvoditi naredbe definirane u glavnoj petlji programa računala.

Autonomno kretanje robota u labirintu i izvršavanje svih potrebnih zadataka odvija se na sljedeći način.

I Teensy i Raspberry imaju svoj program čija je srž „while“ petlja koje se vrti „beskonačno mnogo“ puta.

Okvirno, jedno izvršavanje while petlje Raspberry Pi računala se sastoji od naredbi;

- Provjeri da li je stigao odgovarajući string s serijskog kanala Teensy-ja
- Ukoliko je stigao odgovarajući string s Teensy-ja izvrši određenu funkciju za prepoznani string
- Ukoliko nije stigao odgovarajući string s Teensy-ja, slikaj sliku, vidi da li se na slici nalazi željeno slovo ili boja
- Ukoliko se na slici nalazi željeno slovo ili boja, pošalji string Teensy-ju preko serijskog kanala komunikacije, te će Teensy izvršiti predviđenu funkciju za dati string

Ovo je slijed naredbi koji se ponavlja „beskonačno“. Najbitnija stvar u petlji je ta da cijelo vrijeme Raspberry Pi i Teensy „oslušuju“ jedan drugoga i izvršavaju određene funkcije kada dobiju određenu naredbu.

Struktura glavne petlje mikrokontrolera logički je identična strukturi glavne petlje računala, samo se ne radi o funkcijama za prepoznavanje slike već o kretanju robota ovisno o očitanju senzora.

Također glavna beskonačna petlja mikrokontrolera podijeljena je na manje petlje koje služe kao P regulatori praćenja orijentacije, poravnavanja i držanja željene udaljenosti od zidova i sl.

Cijeli program mikrokontrolera bazira se na tome da robot ide naprijed sve dok ne naiđe na zid sa svoje prednje strane. Ukoliko naiđe na zid sa svoje prednje strane pomoću senzora udaljenosti određuje u kojem smjeru je potrebno skrenuti, s tim da uvijek prati lijevi zid, jer nam ova naizgled jednostavan algoritam omogućava istraživanje velikog udjela uobičajenih labirinta koji se dolaze kao primjer na raznim natjecanjima ovakvog ili sličnog tipa, ali i ima naravno svoje nedostatke, kao što je mogućnost kretanja u „krug“ kao i za određene strukture labirinta nemogućnost potpunog istraživanja labirinta ukoliko se labirint sastoji od otočja koja nisu povezana s rubnim zidom labirinta. Također u međuvremenu dok robot ne naiđe na zid sa svoje prednje strane robot pomoću „if“ naredbe grananja provjerava da li je naišao na crno ili reflektivno sivo polje, temperaturnu žrtvu, nagib i sl. Budući da robot treba veću kinetičku energiju da savlada prepreku za usporavanje (eng. Speed bumps) ili uzbrdicu od 25° potrebno mu je automatski povećati brzinu kretanja, što je realizirano P-regulatorom odnosno očitavanjem z-osi žiroskopa te ukoliko je nagib robota veći od određene vrijednosti do koje toleriramo nagib (npr. 10°) za svaki naredni stupanj nagiba preko 10° regulator povećava brzinu robota za određeni iznos, što rezultira uspješnijim savladavanjima graničnika brzine i polja koja su pod određenim nagibom u labirintu. Također, uz ovaj P-regulator na robotu su prisutna još dva P-regulatora.

Prvi regulator služi za regulaciju praćenja definirane orijentacije na način da za veća odstupanja od željene orijentacije realizira veću brzinu zakretanja prema željenoj orijentaciji, dok drugi regulator služi da spriječi robota da se u gibanju prema naprijed previše približi lijevom ili desnom zidu. Regulator vrši zakretanje robota u lijevo ako se nalazi blizu desnom zidu tj. zakretanje u desno da se robot nalazi blizu lijevom zidu. Brzina zakretanja je veća što je robot bliže lijevom ili desnom zidu jer je veća mogućnost od sudara sa zidom što zapravo i je proporcionalni ili P-regulator. Orijentacija robota pri svakom skretanju se određuje pomoću poravnavanja robota s nekim od zidova koji je trenutno detektiran. Ukoliko postavljena orijentacija robota nije idealna, a uglavnom neće biti idealna, navedeni regulatori će korigirati nesavršenosti te samim tim povećati programsku robusnog robota.

4.6.1 Obrada slike

Postupak obrade slike podijeljen je u dvije faze:

- Prepoznavanje boje
- Prepoznavanje slova

Postupak objašnjavanja započnimo analizom postupka prepoznavanja boje. Prva stvar koju je potrebno učiniti je uslikati sliku na kojoj se nalazi određena obojana žrtva u npr. zelenu boju. U Python programskom jeziku postoje razni tipovi filtera koji omogućuju prepoznavanje određene boje sa slike. Prvi problem na koji nailazimo je taj da ne znamo koje su temeljne značajke boje koju želimo prepoznati odnosno njene temeljne značajke koje jednoznačno

određuju o kojoj se nijansi boje zapravo radi. Temeljne značajke koje jednoznačno određuju o kojoj se boji i nijansi boje zapravo radi mogu biti prikazane u različitim formatima kao što su RGB (eng. Red, Green, Blue), HSV (eng. Hue, Saturation, Value) i sl.

Kao prvi korak izrađena je python skripta pomoću koje klikom na određeni piksel uslikane žrtve dobijemo ispis vrijednosti boje piksela u i HSV formatu. Budući da se u projektu uglavnom koristi HSV format, potrebno je izvršiti skaliranje HSV vrijednosti koje zbog svoje strukture izgledaju vrlo neujednačeno, u određeni interval. Odabrani interval je u rasponu od 0-255 za H, S i V vrijednost pojedine boje.



```

Python 3.7.3 (/usr/bin/python3)
>>> %cd /home/pi/Desktop/v.4.2_ImageRecognition/2.Step
>>> %Run 2.clickFindHSV.py

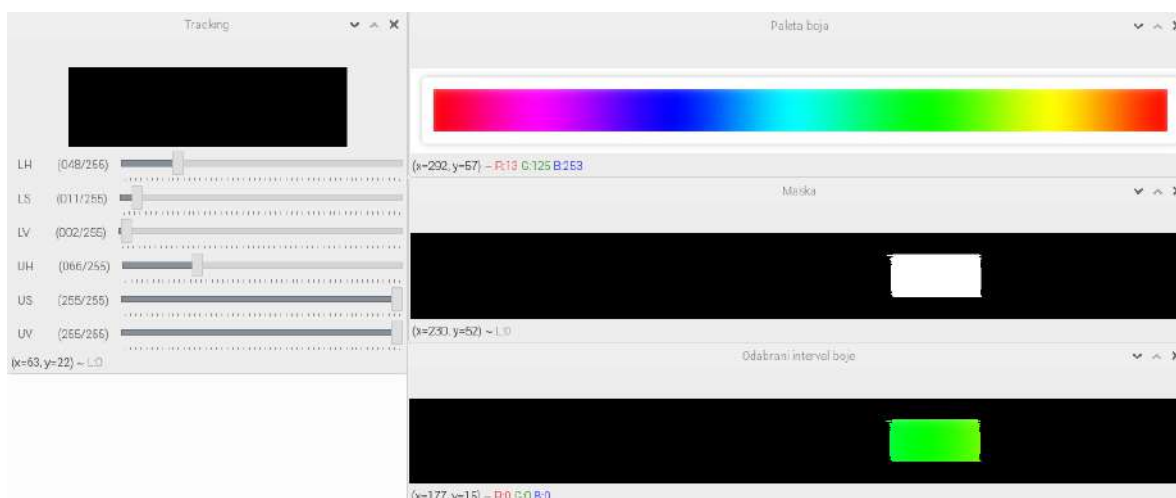
Enter 'r' for red image, 'y' for yellow image or 'g' for green image.
g
Dimensions of image are:
width 1920
height 1080

RGB    71    64    49
HSV    40.9   0.31   0.278
HSV_S  29.0   79.0   71.0

```

Slika 4.55 Ispis HSV vrijednosti odabranog piksela

Kada je jednoznačno određena vrijednost boje koju percipira kamera robota, potrebno je proširiti interval vrijednosti kako bi povećali robusnost algoritma, budući da zelenu boju ne predstavlja samo jedna vrijednost, nego cijeli interval nijansi, stoga je potrebno izraditi novu skriptu pomoću koje će se moći odabrati određeni interval za zelenu, žutu i zelenu boju. Ova skripta izrađena je na način da je u nju učitana slika palete svih mogućih boja, te pomjeranjem klizača sučelja skripte dinamički mijenjamo minimalne i maksimalne vrijednosti intervala boja u HSV formatu, što rezultira kreiranjem maske odnosno filtera boje na sliku koja prikazuje cijelu paletu boja, odnosno dolazi do dinamičkog uklanjanja svih boja koje se nalaze izvan definiranog intervala te dinamičkog prikazivanja odabranog rezultata boja na prozoru sučelja skripte. Prikaz sučelja ove python skripte dan je u nastavku.



Slika 4.56 Prikaz sučelja Python skripte za određivanje intervala određene boje

Treći i najbitniji korak je realiziranje automatskog prepoznavanja boja sa slike, a realizirat ćemo ga na način da se na sliku koju kamera robota uslika, stave filtri intervala boje koju smo definirali u prethodnom koraku za crvenu, žutu i zelenu boju. Kada dođe do preklapanja boja izvorne slike s odabranim filterom, na piksele koji su unutar odabranog intervala postaviti ćemo masku koja je bijele boje, dok piksele izvan maske postavljamo u crnu boju. Nakon što je slika postala crno bijela, preostalo je samo prebrojiti bijele piksele slike što omogućava Open CV naredba „cv2.countNonZero()“. Kada smo prebrojali bijele piksele, potrebno je odrediti dimenzije slike što također omogućava Open CV modul, te jednostavnom matematičkom radnjom dijeljenja broja bijelih piksela slike s brojem svih piksela slike odrediti koliki postotak slike pokriva određena boja. Kako bi se spriječile smetnje, potrebno je postaviti interval koji npr. kaže ako se na slici nalazi više od 3% i manje od 90% crvene boje tada je prepoznana crvena žrtva. Minimalni interval postavljen je iz razloga kako bi se spriječio slučaj, ako program prepozna jako mali broj crvenih piksela, npr. samo jedan piksel crvene boje na slici dimenzija 1920x1080 koja ukupno ima više od 2 milijuna piksela, točnije 2 073 600 piksela, očigledno je da se radi o šumu kojeg treba eliminirati, također ako se desi da program javlja 2 073 600 crvenih piksela također se radi o šumu jer razne refleksije u neidealnim uvjetima uzrokuju promjenu jednoznačne vrijednosti boje (djelomična sjena na meti i sl.). Navedeni postupak potrebno je provesti za svaki interval boje pojedinačno.

Nakon izrade algoritma za prepoznavanje boje potrebno je realizirati prepoznavanje slova sa slike.

Prepoznavanje slova sa slike realizirat ćemo pomoću OCR (eng. Optical Character Recognition) Tesseract modula koji je podržan i od strane Raspberry Pi računala kao modul naziva Pytesseract. Budući da se radi o gotovom OCR modulu, nije potrebno prikupljati bazu podataka i definirati niz pravila kako bismo prepoznali određeno slovo što bi bilo potrebno ukoliko bismo išli sa treniranjem vlastitog modela konvolucijskih neuralnih mreža što predstavlja količinu informacija dovoljnu za zaseban diplomski rad. Budući da je prema Robocup natjecanju i jednostavnost stavka koja se boduje na određeni način, odlučili samo se za korištenje navedenog pytesseract modula.

Postupak instaliranja ovog modula je kao i postupak instaliranja svih drugih modula u python programskom jeziku. Poželjno je da se postupak prepoznavanja slova iz slike odvija na najbrži mogući način, stoga je potrebno pytesseract modul konfigurirati na najprihvatljiviji način. Mogući režimi rada kao i određene ključne riječi ovog modula su navedeni u nastavku.

```

Usage:
  tesseract --help | --help-extra | --help-psm | --help-oem | --version
  tesseract --list-langs [--tessdata-dir PATH]
  tesseract --print-parameters [options...] [configfile...]
  tesseract imagename|imagelist|stdin outputbase|stdout [options...]
[configfile...]
OCR options:
  --tessdata-dir PATH    Specify the location of tessdata path.
  --user-words PATH      Specify the location of user words file.
  --user-patterns PATH   Specify the location of user patterns file.
  -l LANG[+LANG]         Specify language(s) used for OCR.
  -c VAR=VALUE           Set value for config variables.
                        Multiple -c arguments are allowed.
  --psm NUM              Specify page segmentation mode.
  --oem NUM              Specify OCR Engine mode.
NOTE: These options must occur before any configfile.
Page segmentation modes:
  0  Orientation and script detection (OSD) only.
  1  Automatic page segmentation with OSD.
  2  Automatic page segmentation, but no OSD, or OCR.
  3  Fully automatic page segmentation, but no OSD. (Default)
  4  Assume a single column of text of variable sizes.
  5  Assume a single uniform block of vertically aligned text.
  6  Assume a single uniform block of text.
  7  Treat the image as a single text line.
  8  Treat the image as a single word.
  9  Treat the image as a single word in a circle.
 10  Treat the image as a single character.
 11  Sparse text. Find as much text as possible in no particular order.
 12  Sparse text with OSD.
 13  Raw line. Treat the image as a single text line,
    bypassing hacks that are Tesseract-specific.
OCR Engine modes: (see https://github.com/tesseract-ocr/tesseract/wiki#linux)
  0  Legacy engine only.
  1  Neural nets LSTM engine only.
  2  Legacy + LSTM engines.
  3  Default, based on what is available.
Single options:
  -h, --help             Show minimal help message.
  --help-extra           Show extra help for advanced users.
  --help-psm            Show page segmentation modes.
  --help-oem            Show OCR Engine modes.
  -v, --version          Show version information.
  --list-langs           List available languages for tesseract engine.
  --print-parameters     Print tesseract parameters.

```

Slika 4.57 Prikaz mogućih konfiguracija Tesseract modula [27]

Za ovaj projekt odabran je standardni OCR način pod brojem tri odnosno „Default, based on what is available.“, koji kombiniramo s načinom segmentacije koji se nalazi pod brojem 6 na

prethodnoj slici odnosno „Assume a single uniform block of text.“ iz razloga jer u testu upravo ova kombinacija daje najtočnije i najbrže rezultate.

Navedeno konfiguriranje obavlja se u python skripti zaduženoj za autonomno prepoznavanje boja i slova na način da unesemo sljedeću liniju koda.

```
custom_config = r'--oem 3--psm 6'
```

Prepoznavanje teksta sa slike vrši se pomoću jedne jedine linije koda u kojoj pytesseract funkciji kojoj uz postavke konfiguracije pytesseract modula proslijedimo sliku s koje želimo prepoznati slovo, a funkcija nam vraća što je na slici prepoznano.

Prikaz pomenute naredbe je dan u nastavku.

```
text = tess.image_to_string(obrađena slika, config=custom_config)
```

Neobrađena slika, odnosno slika koja je uslikana kamerom uglavnom ima jako puno smetnji koje treba filtrirati kako bi navedeni modul lakše prepoznao znak koji se nalazi na slici.



Slika 4.58 Prikaz neobrađene slike objekta prikazanog slovom „S“

Prvi korak je izvornu sliku koja se sastoji od tri kanala boja (npr. RGB) pretvoriti u jednokanalnu sliku kako bi sam proces obrade slike bio višestruko jednostavniji. Ovaj korak realiziramo na način da sliku u boji prevedemo u crno-bijelu sliku (eng. Grayscale filter) na kojoj će svaka boja biti određena nijansa „sive“ boje. Kada je slika prevedena u crno bijeli oblik, cilj je što je više moguće ispraviti nesavršenosti koje mogu biti npr. deformirani rubovi, relativno male primjese druge boje u okolini druge boje i sl.



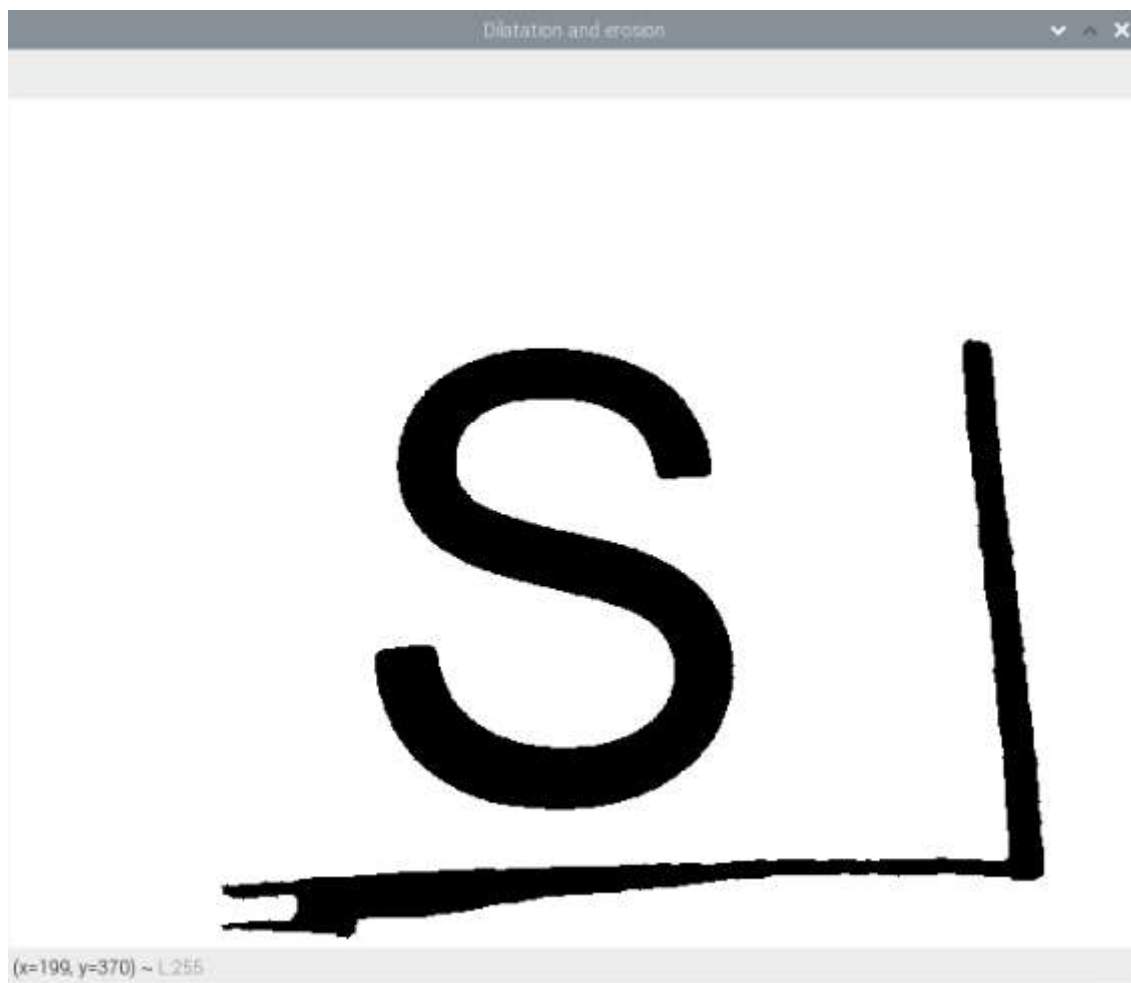
Slika 4.59 Prikaz slike objekta prikazanog slovom „S“ nakon primijenjenog „Grayscale“ filtera

Budući da su slova koja želimo prepoznati sa slike crne boje i nalaze se na bijeloj pozadini, potrebno je se riješiti svih nijansi sive boje, odnosno želimo dobiti sliku koja se sastoji samo od crne i bijele boje. Ova mogućnost se realizira korištenjem „threshold“ filtera koji vrijednosti boje piksela koji su ispod „threshold“ vrijednosti pretvara u crne piksele, a sve vrijednosti boje piksela koje su iznad određene „threshold“ vrijednosti pretvara u bijele piksele.



Slika 4.60 Prikaz slike objekta prikazanog slovom „S“ nakon primijenjenog „Treshold“ filtera

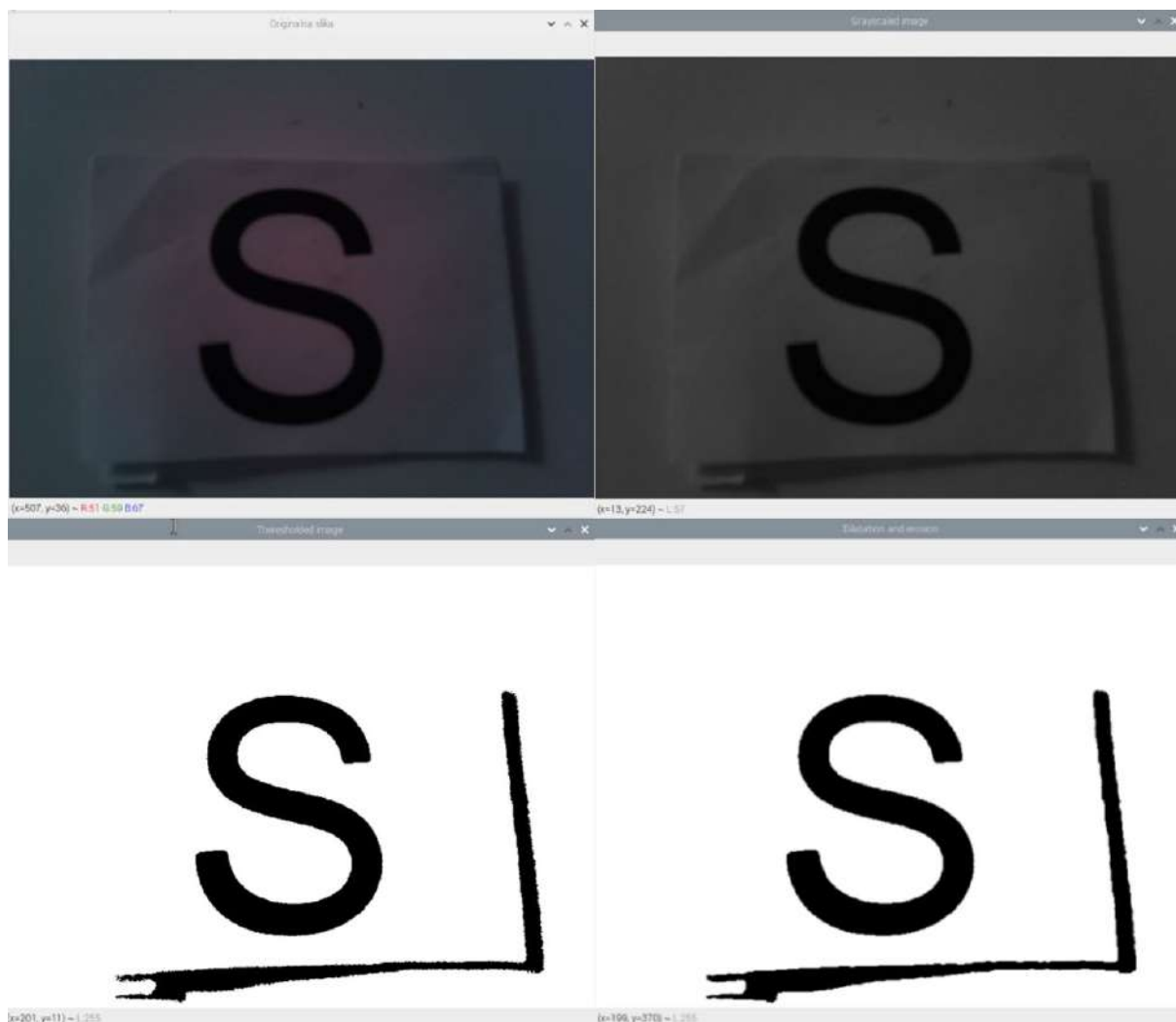
Kada se slika sastoji samo od crne i bijele boje, i vrlo je jasno izdvojen željeni znak kojeg je potrebno prepoznati. Rubove znaka koji su deformirani potrebno je isprofilirati na način da deformacija rubova znaka bude minimalna, a preostale „pikselske mrlje“, odnosno manju skupinu piksela jedne boje koja se nalazi u većinskom okruženju druge boje ukloniti sa slike. Navedeni rezultat postizemo korištenjem filtera erozije i dilatacije (eng. Erosion and Dilataion filter).



Slika 4.61 Prikaz slike objekta prikazanog slovom „S“ nakon primijenjenog „Erosion and Dilataion“ filtera

Uz navedene postoji još mnogo filtera slike koji se ovisno o potrebi mogu primijeniti u postupku predprocesiranja slike koji nama nisu bili potrebni.

Nakon što je početna slika predprocesirana, možemo ju proslijediti pytesseract funkciji koja će uvelike jednostavnije izdvojiti znak koji želimo prepoznati iz slike, što rezultira izrazitom povećanju u točnosti algoritma te povećanju brzine izvođenja algoritma.



Slika 4.62 Prikaz cjelokupne obrade slike

4.7 Mehanička rješenja

4.7.1 Izrada kotača

Pri izradi kotača potrebno je ostvariti određeno prijanjanje kotača uz podlogu kako bi mogli savladati prepreke koje se postavljaju pred robota (graničnici brzine, polja pod nagibom od 25°). Prijanjanje kotača uz podlogu mijenja se promjenom širine kotača kao i odabirom različitih materijala od kojih je izrađen sam kontaktni dio kotača s podlogom.

Kako bi povećali tlak dodirne površine potrebno je odabrati što je manju moguću širinu kotača, koja uz povećanja tlaka dodirne površine kotača i podloge, garantira i pouzdanu funkcionalnost u manevrima koje robot treba obavljati.

Širina kotača za koju smo se odlučili je 7 mm, dok promjer iznosi 50 mm.

Budući da kotač želimo sami proizvesti odnosno postupkom 3D printanja postavlja se pitanje na koji način konstruirati dodirnu površinu kotača s podlogom. 3D model kotača prikazan je na narednoj slici.



Slika 4.63 Prikaz 3D modela kotača

Budući da je trenje između PLA materijala i plastificiranih podloga od kojih će se sastojati labirint vrlo malo, odlučili samo izraditi kotač na način da ćemo prostor između dvaju plavih koncentričnih krugova naznačenih na prethodnoj slici ispuniti kombinacijom smjese gume u tekućem stanju, proizvođača DWR Plastic, kako bi se, nakon što smjesa očvrsne, između navedenih plavih krugova dobio mekan gumeni volumen čije je trenje dodirne površine s podlogom znatno veće od trenja PLA materijala s dodirnom površinom podloge labirinta.

Smijese miješamo u omjeru 1:1, na način da izbjegnemo stvaranje mjehurića u smjesi iz razloga jer će upravo ti mjehurići u konačnoj smjesi činiti prelazan prostor, što svakako nije poželjno. Nakon što je smjesa izmiješana i ulivena u prostor između dviju kružnica s prethodne slike, potrebno je pričekati u vremenskom periodu od oko 120 minuta da smjesa postane kompaktna. Kada je smjesa kompaktna, mehaničkim putem uklanjamo volumen koji se nalazi ispod vanjskog plavog kruga, dok smjesa ostaje kompaktno vezana za nazube koji su pričvršćeni za volumen unutarnjeg plavog kruga za slike.

Korištene smjese su DWR Plastic 1-2-1 Easy Mould Part A Soft i DWR Plastic 1-2-1 Easy Mould Part B Soft. Prikaz ambalaža navedenih smjesa dan je na sljedećoj slici.



Slika 4.64 Prikaz smjese gume korištenih u projektu [28]

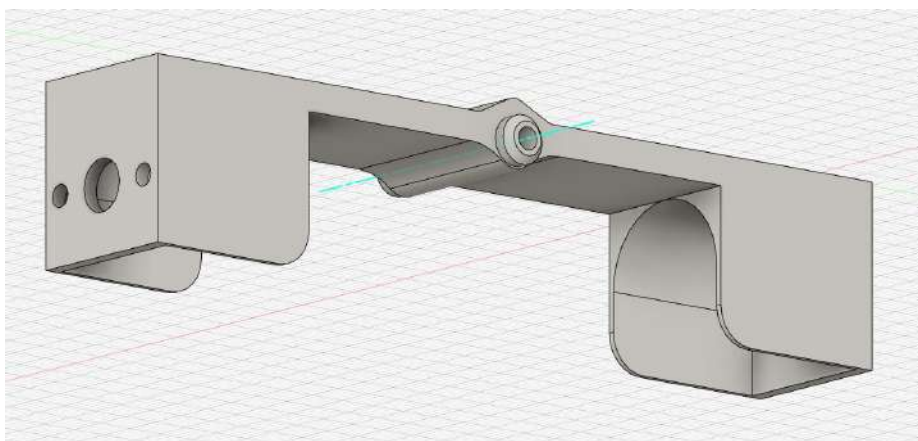
Izrađene kotače potrebno je pričvrstiti za izlaznu osovinu DC motora stoga je potrebno korištenje adaptera Pololu Universal Aluminum Mounting Hub za izlaznu osovinu DC motora promjera 4mm koji je prikazan u nastavku zajedno s primjerom pričvršćivanja s izlaznom osovinom DC motora i proizvoljnog kotača.



Slika 4.65 Prikaz montiranja adaptera za izlaznu osovinu DC motora i kotača [29]

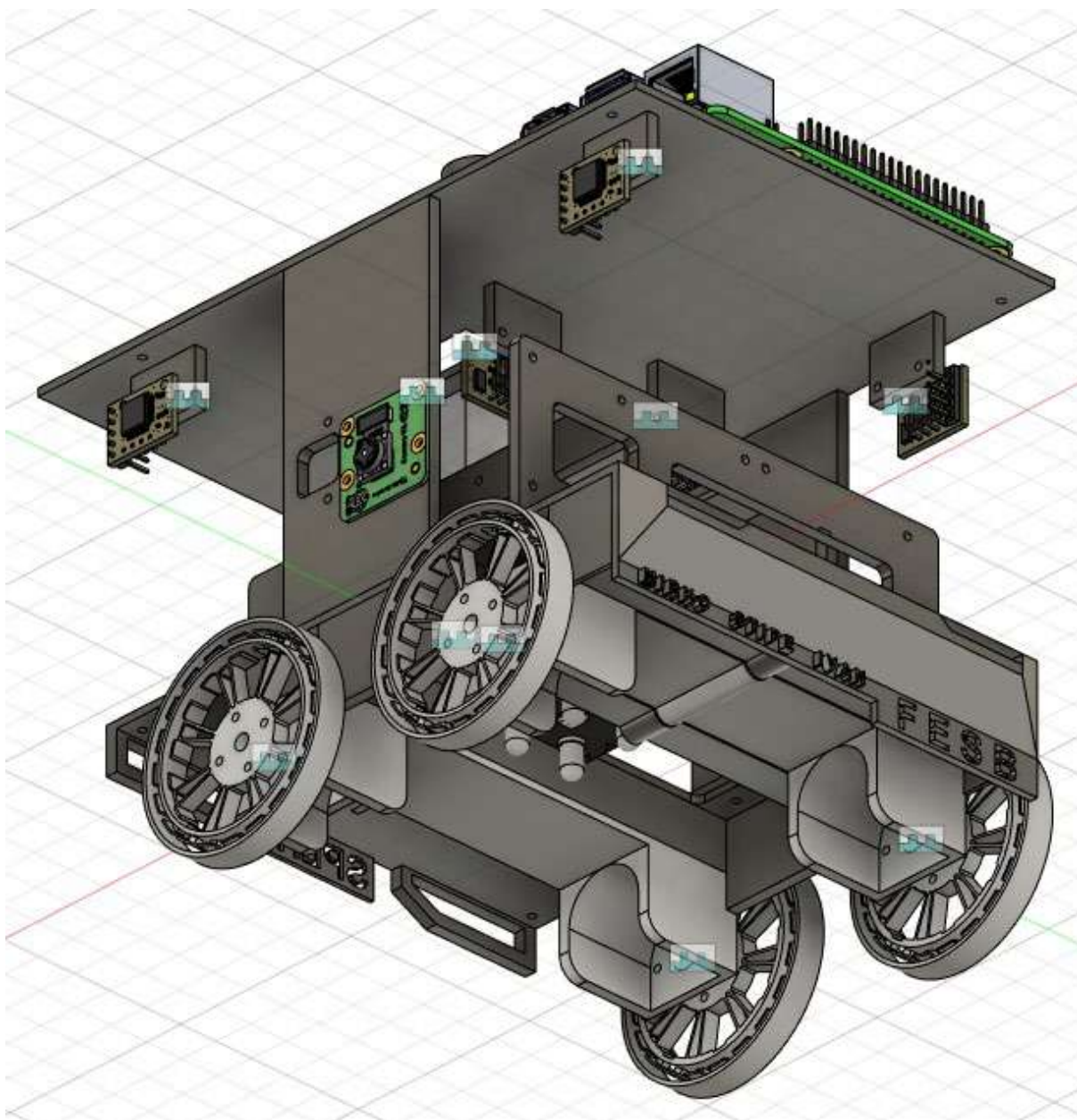
4.7.2 Izrada amortizirajuće osovine

Budući da u određenim situacijama u kojima se može naći robot u labirintu dolazi do lebdenja kotača u odnosu na podlogu, a samim tim povećava se mogućnost da robot jednostavno zapne na nekoj od kombinacija krhotina, graničnika brzine i polja labirinta koja se nalaze pod nagibom, stoga je potrebno razviti amortizirajuću prednju osovinu koja se može rotirati za određeni kut koji je dovoljan da se amortiziraju i najveće prepreke u labirintu. Prikaz 3D modela amortizirajuće osovine na koju se montiraju prednji DC motori dan je na sljedećoj slici.



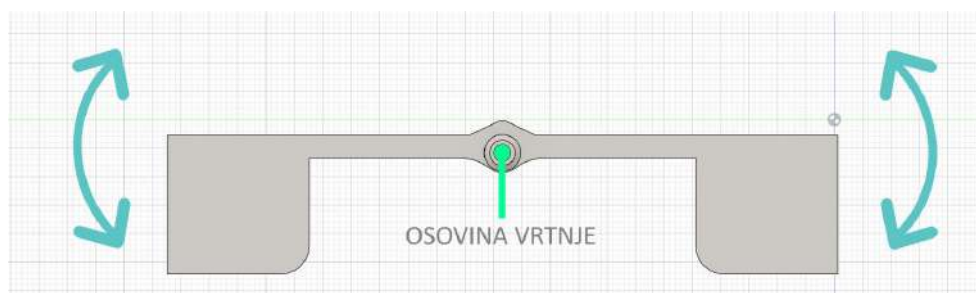
Slika 4.66 3D model prednje amortizirajuće osovine

Prikaz prednje amortizirajuće osovine integrirane u tijelo robota dan je na sljedećoj slici.



Slika 4.67 Prikaz 3D modela amortizirajuće osovine integrirane u tijelo robota.

Smjerovi zakretanja oko rotirajuće osi amortizirajuće osovine prikazani su na sljedećoj slici.

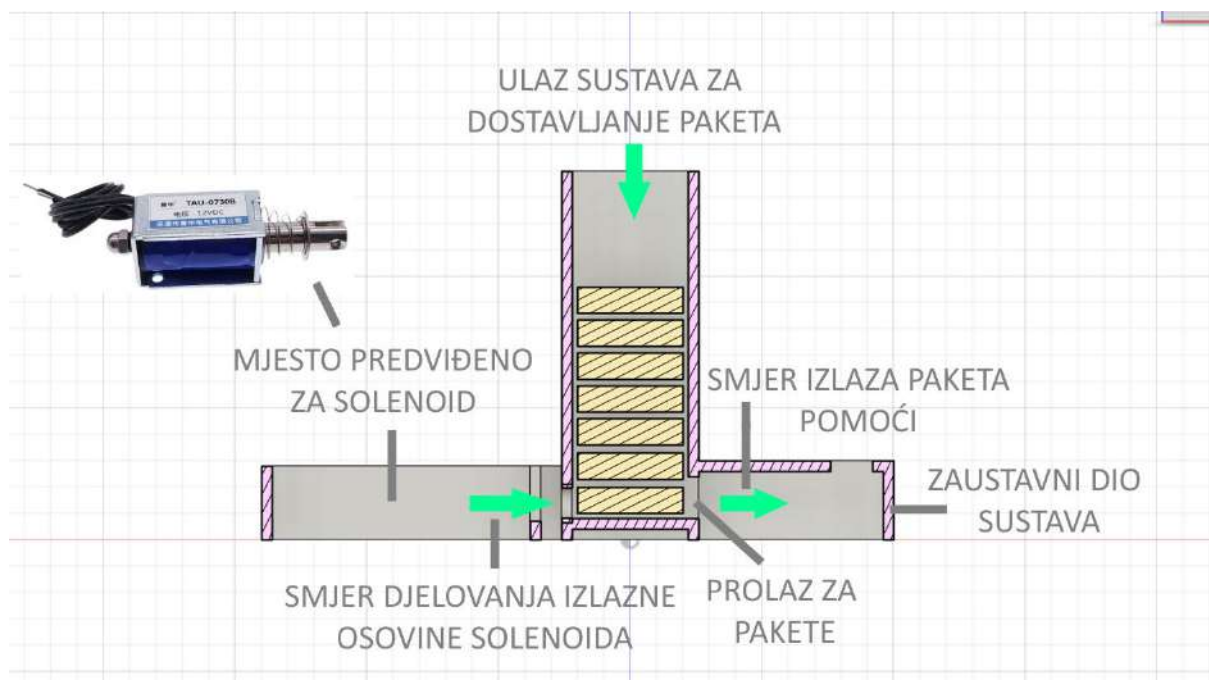


Slika 4.68 Prikaz osovine i smjerova zakretanja amortizirajuće osovine

Još robusnija izvedba robota bi bila s dvije amortizirajuće osovine, ali zbog jednostavnije izvedbe i impresivnih rezultata manevriranja robotom kroz moguće prepreke zahvaljujući prednjoj amortizirajućoj osovini zaključeno je da trenutno nema potrebe za navedenim unaprjeđenjem, čemu u prilog ide i pogon robota na sve kotače tako da će u najgorem slučaju robot biti pogonjen s tri kotača vezana za podlogu što je dovoljno za uspješno manevriranje labirintom.

4.7.3 Sustav za dostavljanje paketa pomoći

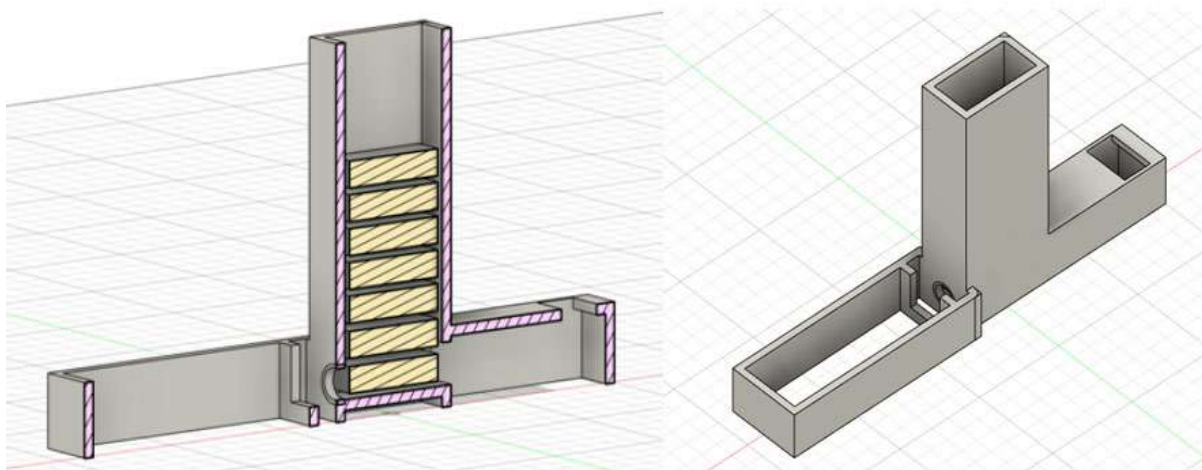
Presjek 3D modela sustava za dostavljanje paketa pomoći prikazan je na sljedećoj slici.



Slika 4.69 Prikaz funkcionalnosti sustava za dostavljanje paketa pomoći

Paketi pomoći u sustav se ubacuju kroz ulaz ovog mehanizma prikazanog na prethodnoj slici.

Paketi su složeni jedan iznad drugog kako bi bila iskorištena sila Zemljine teže u funkcioniranju cijelog mehanizma. Pomoću izlazne osovine solenoida gura se najdonji paket pomoći kroz prolaz za pakete čije su dimenzije dovoljne da kroz njega prođe samo jedan paket pomoći. Nakon što je najdonji paket prošao kroz prolaz, paket koji je bio iznad njega zbog utjecaja sile Zemljine teže sada dolazi na mjesto izguranog paketa, a izgurani paket, s obzirom da posjeduje relativno veliku brzinu, zaustavlja se pomoću zaustavnog dijela sustava nakon čega pod utjecajem sile Zemljine teže izlazi iz sustava. Zaustavni dio sustava uvelike povećava preciznost dostavljanja paketa pomoći.



Slika 4.70 Prikaz 3D modela sustava za dostavljanje paketa pomoći

5 EKSPERIMENTALNA ISTRAŽIVANJA

Prepoznavanje meta u obliku slova i boje:

Uspješnost prepoznavanja i kategorizacije objekata predstavljenih slovom i bojom koji su adekvatno osvijetljeni te su u stacionarnom stanju u odnosu na optičku kameru iznosi 97% na uzorku od 100 pokušaja. Ukoliko je meta neadekvatno osvijetljena (prisutne su razne refleksije, nedostatak dovoljne količine svjetlosti) robot nije u mogućnosti prepoznati objekte s pouzdanom sigurnošću.

Ukoliko se robot kreće u odnosu na metu, algoritam ne stigne optičkom kamerom pokriti prazninu koja nastaje zbog vremena koje zahtjeva funkcija za prepoznavanje slova kojoj je potrebno prosječno 700 ms da obradi uslikanu sliku. Potrebno izraditi algoritam kojem je za prepoznavanje slova sa slike potrebno okvirno 100 ms, što je moguće realizirati pomoću konvolucijske neuralne mreže.

Brzina algoritma za prepoznavanje boje je prosječno 25 ms, što je više nego dovoljno brzo za zahtjeve natjecanja.

Potrebna udaljenost senzora optičke kamere da bi uslikala objekt 4 cm x 4 cm koji je u idealnoj razini kamere je 10 cm, što nije jednostavno realizirati u labirintu čija su polja dimenzija 30 cm x 30 cm. Stoga je potrebno koristiti kameru s širim vidnim poljem kako bi kamera robota bila u stanju fotografirati slike koje nisu u idealnoj razini s kamerom s udaljenosti od oko 6 cm ili manje, ili pak optičku kameru premjestiti na položaj bliži središtu robota.

Prepoznavanje temperaturne mete:

Potrebno kalibrirati postojeće MRMS senzore kako bismo znali koja analogna vrijednost predstavlja određenu temperaturu ili pak koristiti Adafruit AMG8833 IR termalnu kameru koja dolazi kalibrirana te je znatno podržanija literaturom od termalne kamere korištene u ovom projektu.

Mehanizam dostavljanja paketa pomoći:

Na uzorku od 100 pokušaja uspješnost mehanizma za dostavljanje paketa pomoći je 90%, što se može znatno povećati minimalnim modifikacijama izgleda paketa pomoći koji u 10% slučajeva zapne iz razloga rubova koji ostaju na paketu nakon odvajanja isprintanog 3D modela od prvog sloja printanja. Još jedno od rješenja je povećati toleranciju mehanizma za dostavljanje paketa pomoći u odnosu na dimenzije paketa pomoći, te ugradnja sloja amortizirajućeg materijala (spužve) na zaustavni dio sustava za dostavljanje paketa pomoći kako bi kinetička energija izlaznog paketa pomoći bila znatno više apsorbirana, što rezultira povećanjem preciznosti mehanizma za dostavljanje paketa pomoći.

Očitavanje podataka o orijentaciji robota:

Ovaj podsustav funkcionira odlično, a testiranje je provedeno na uzorku od nekoliko sati. Dozvoljena tolerancija je programski izvedena i iznosi 1° kako bi spriječila neželjene oscilacije sustava.

Senzori udaljenosti:

MRMS lidari su se pokazali kao nepouzdana opcija te su zamjenjeni Sharp GP2Y0A41SK0F. Tolerancija točnosti Sharp senzora udaljenosti je ± 1 cm na intervalu od 4 do 27 cm što je prihvatljivo za navedene kriterije. Pri očitanjima su prisutni povremeni šumovi koji su programski svedeni na zanemarivu razinu.

Prepoznavanje crnih polja:

Točnost ovog postupka je 100% na uzorku od nekoliko stotina pokušaja detektiranja crnih polja.

Regulator izbjegavanja bočnih zidova:

Princip rada odličan. Potrebna dodatna kalibracija za poboljšane performanse. Poželjno uvođenje dva bočna senzora udaljenosti koji s prednjom i bočnim stranama robota zatvaraju kut od 45° .

Regulator brzine ovisno o nagibu oko Z-osi:

Princip rada odličan. Potrebna dodatna kalibracija za poboljšane performanse. Poželjna ugradnja utega na predviđeni prostor iza prednje maske robota kako bi robot bio stabilniji pri usponu uz polje nagiba od 25° , te imao bolje prijanjanje uz podlogu.

Sustav amortizacije:

Radi besprijeckorno za prepreke niže od 1 cm. Potrebno je ugraditi amortizere na amortizirajuću osovinu kako bi robot bio stabilan pri gibanju niz nagib od 25° .

Prianjanje uz podlogu:

Vrlo dobro. Poželjni mekši profili na gumama kako bi se poboljšalo prianjanje u trenucima savladavanja nagiba od 25% . Umjesto smjesa guma „Soft“ koristiti smjesu „Extra Soft“ istog proizvođača.

Autonomija sustava:

Robot je sposoban za djelomičnu autonomnu vožnju labirintom. Potrebna opsežna testiranja svih položaja u kojima se robot može naći kao bi se korigirali postojeći i isprogramirali novi regulatori potrebni za potpunu autonomnu vožnju labirintom. Cjelokupan sustav komunicira na željni način. U slučaju snažnijih udara dolazi do gubitka komunikacije za žiroskopom.

6 ZAKLJUČAK

Robot namijenjen Robo Cup Rescue Maze natjecanju, vrlo je zahtjevan projekt za koji je potrebno izdvojiti jako puno vremena na istraživanje, projektiranje i testiranje rada. Utrošeno vrijeme u projekt je oko godinu i pol dana. Okvirno 35% vremena je potrošeno u istraživanja cjelokupnog principa rada svih sustava i podsustava robota, 30% vremena je potrošeno u 3D projektiranje i izradu tijela robota, 30% vremena je potrošeno na spajanje svih komponenti iz razloga što su se mnogi izgledi robota mijenjali, a komponente su ostajale iste. Kada se radi s jednom komponentom šanse za uspjeh su vrlo velike, dok s dodavanjem svake nove komponente šanse za uspjehom eksponencijalno opadaju. Na robotu postoji više od 50 hardverskih funkcija koje svaka zasebno moraju biti testirane i funkcionalne. Svega 5% vremena je provedeno na testiranje regulatora i naprednijih funkcija koje omogućavaju autonomiju robota. Ovaj mali postotak testiranja robota proizlazi iz same prirode projekta u kojem je polazno potrebno povezati hardverske komponente na način da su u mogućnosti ispuniti određene kriterije. Nakon što je cjelokupan hardver fizički i logički funkcionalan može se krenuti s opsežnim testiranjima cjelokupnog (hardversko-softverskog) sustava. Procjena je da je potrebno barem godinu dana intenzivnog rada jedne osobe na testiranju i unaprjeđivanju funkcionalnosti robota kako bi se ispunili kriteriji navedeni u pravilniku iz uvoda.

Potrebne nadogradnje robota pobrojane su u nastavku:

- Povećati brzinu izvršavanja algoritma za prepoznavanje slova
- Realizirati osvjetljenje koje osvjetljava prostor koji snimaju optičke kamere
- Realizirati mapiranje prostora
- Povezati izvode s računala, mikrokontrolera s korištenim perifernim uređajima na robusniji i jednostavniji način.
- Potrebna kamera koja omogućava širi kut gledanje FOV
- Potrebna ugradnja amortizera na amortizirajuću osovinu
- Potrebna ugradnja dva para senzora za mjerenje udaljenosti
- Ugradnja ekrana

U nastavku su navedeni razlozi zbog kojih je potrebna nadogradnja kao i općeniti način realizacije pojedine nadogradnje.

Algoritam za prepoznavanje slova sa slike izvodi se brzinom koja je na granici prihvatljivosti, odnosno da bi se slovo izdvojilo s jedne slike potrebno je prosječno 700 ms, dok je 50 ms vremena potrebno da se izvrše svih preostalih funkcija algoritma koji se vrti na Raspberry Pi računalu (okidanje slike, filtriranje slike, prepoznavanje boja...) Na standardnim više-jezgrenim računalima postoji mogućnost da se na svaku jezgru procesora računala šalje određeni segment funkcije za prepoznavanje slova iz slike, što još uvijek nije podržano za Raspberry Pi računalu, što će vjerojatno biti podržano kroz određeno vrijeme. Također ovaj problem može se fiktivno riješiti na način da se uspori kretanje robota i to korištenjem drugačijeg prijenosa na DC motorima koji omogućava pravljenje manjih koraka izlazne osovine na kojoj se nalazi kotač. Također je važno napomenuti da ovo nije stvarno rješenje jer utječe na

brzinu obavljanja zadatka što je jedan od prioriternih kriterija za spašavanje stvarnih i fiktivnih žrtvi. Drugo rješenje je prepoznavanje znakova sa slike pokušati realizirati korištenjem drugih modula pomoću kojih je moguće prepoznavanje slova sa slike (npr. Tensorflow). Također neke od opcija su korištenje utrenirane ili pak treniranje vlastite konvolucijske neuralne mreže prilagođene zahtjevima projekta. Također uz Raspberry Pi koji je korišten o ovom projektu dobro bi se bilo detaljno upoznati s računalima-mikrokontrolerima specifične namjene za prepoznavanje objekata sa slike, a koje proizvodi kompanija Nvidia.

Kako bi utjecaj različitih izvora osvjetljenja na sam proces prepoznavanja boja i slova bio umanjen, idealno bi bilo kada bi robot posjedovao vlastito osvjetljenje prostora kojeg kamera snima kako bi boja prostora koji kamera snima u različitim uvjetima osvjetljena ostala nepromijenjena te ne bi bilo potrebe za kalibriranjem boja kamere svaki put kada se uvjeti osvjetljenja minimalno promjene, što uvelike povećava robusnost sustava zaduženog za prepoznavanje slova i boja iz okoline.

Idealno bi bilo kada bi robot posjedovao mogućnost mapiranja prostora, međutim to je mogućnost koja zahtjeva jako puno kombinacija testiranja i podešavanja određenih softverskih i hardverskih parametara, što zahtjeva ozbiljnu količinu uloženog vremena. Okvirna ideja je na robot ugraditi jedan par enkodera pomoću kojih bi se moglo znati za koliku je se udaljenost robot pomaknuo. Budući da su dimenzije polja labirinta poznate i iznose 30 cm x 30 cm, a robot posjeduje žiroskop, moguće bi bilo na vrlo jednostavan način znati koji je trenutni položaj robota u odnosu na početni položaj koji bi označili koordinatama, odnosno pridruženima parom (0,0). Ako bi oba enkodera koja smo npr. postavili na prednje kotače robota, izmjerili minimalno 30 cm, ovisno o orijentaciji koju žiroskop pokazuje zaključak bi bio da je se robot pomaknuo na novo polje koje se može ovisno o orijentaciji kodirati pridruženim parom (0,1) ili pak (1,0) što je proizvoljna odluka. Kako bi se robot kretao u određenom smjeru tako bi u određeni niz dodavali polja kroz koja je robot prošao. Ovisno o udaljenosti koju pokazuju senzori udaljenosti, moglo bi se detektirati s koje strane polja se nalaze zidovi. Budući da je sve navedeno potrebno na pregledan način povezati ideja je u Arduino programskom jeziku kreirati trodimenzionalni objekt kojem će prva dimenzija biti orijentacija po x-osi određenog koordinatnog sustava, druga dimenzija će biti orijentacija po y-osi određenog koordinatnog sustava dok bi treća dimenzija objekta bio niz stringova pomoću kojih bi kodirali prepoznane žrtve, crna i reflektirajuće siva polja kao i položaje zidova za određeno polje. Na ovaj način bi vrlo jednostavno bilo omogućeno softversko pretraživanje polja u dvodimenzionalnom koordinatnom sustavu, te iščitavanje određene posebnosti određenog polja pomoću kojih bi npr. mogli spriječiti dostavljanje paketa pomoći žrtvi kojoj je već dostavljena pomoć, ali smo igrom slučaja na nju ponovno naišli, te ostali bez paketa pomoći koji je potrebno udijeliti nekoj od preostalih žrtvi. Navedene podatke bi ovisno o prepoznavanju „checkpoint“ polja upisivali u trajnu memoriju mikrokontrolera ili Raspberry Pi računala, te nakon što su sve žrtve identificirane, sve potrebne podatke poslati python skripti koja ima sposobnost pretraživanja polja labirinta, odnosno niza, kako bi robotu dala upute o prolasku najkraćim putem od trenutnog konačnog položaja do početnog polja labirinta s kojeg je robot krenuo. Python skripta koja bi omogućavala navedeno može biti izrađena pomoću python modula „Turtle graphics“ koji uz vizualizaciju rasporeda polja labirinta s određenim značajkama nudi mogućnost

pretraživanja navedenih polja pomoću raznih algoritama pretraživanja strojnog učenja (npr. A* algoritam), kako bi najkraćim mogućim putem robot dospio na početno polje.

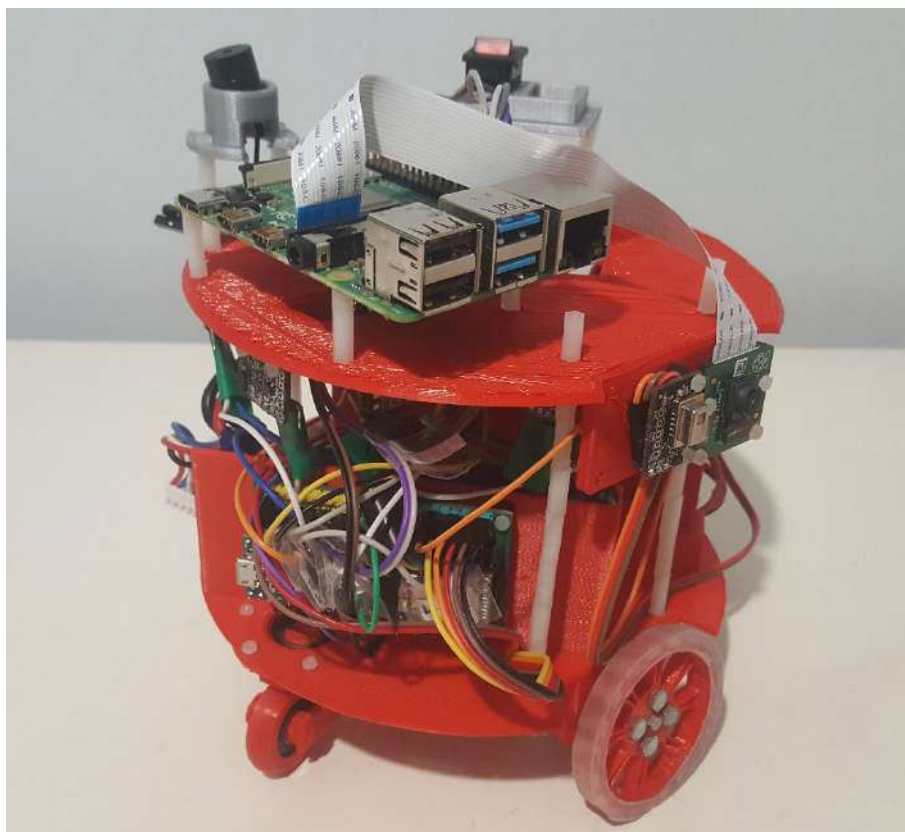
Na robotu se nalazi jako puno kabela što uvelike otežava preglednost i mogućnost pristupa pojedinim pinovima mikro kontrolera. Rješenje je moguće izradom vlastite PDB pločice uz premještanje mikro kontrolera Teensy 3.5 na gornju konstrukciju robota.

Optička kamera treba biti od objekta udaljena 10 cm kako bi uslikala cjelovit objekt 4cm x 4cm koji je idealnoj razini kamere. Ukoliko dođe do pomjeranja objekta više od 1 cm gore ili dole, objekt će izaći iz vidnog polja kamere, stoga je potrebno koristiti kameru koje ima šire vidno polje (eng. Field Of View). Jedna od opcija je Raspberry Pi HQ kamera.

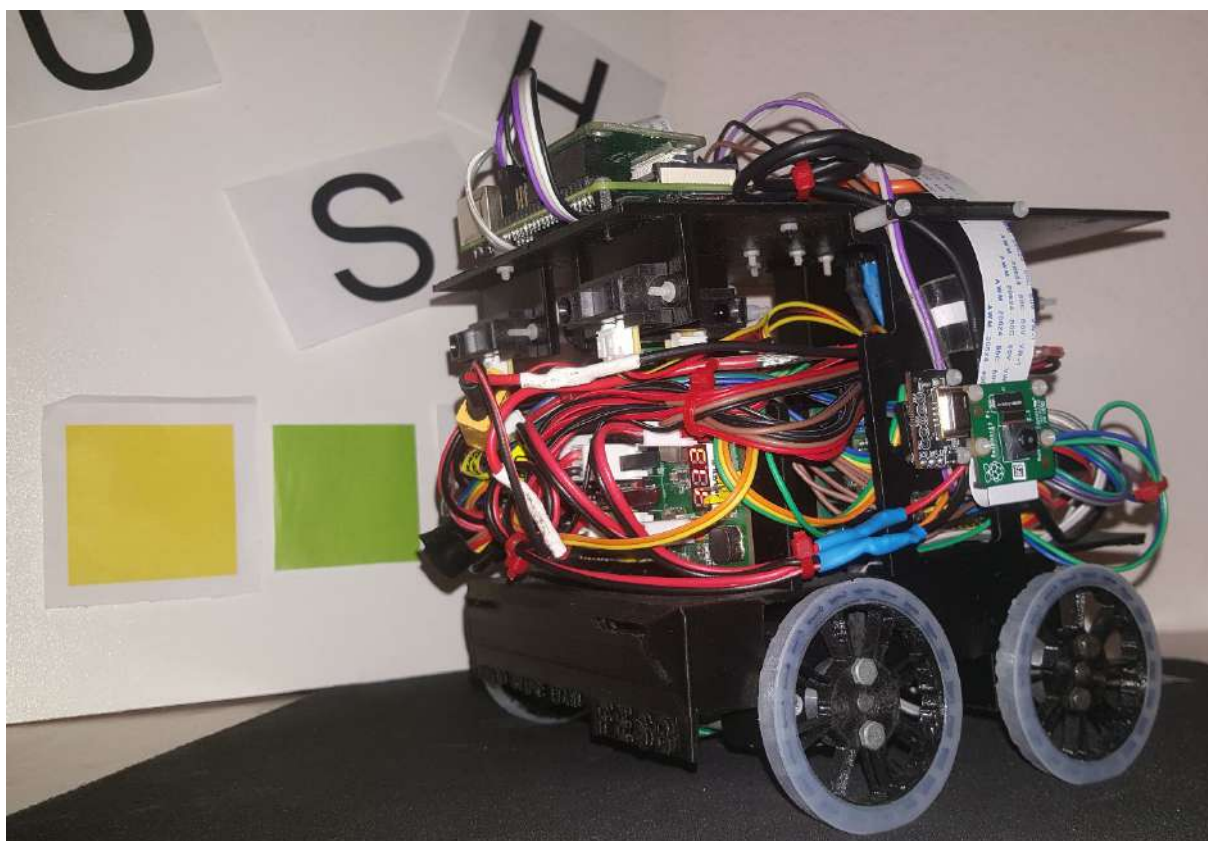
Potrebno je ugraditi amortizere na amortizirajuću osovinu kako bi robot bio stabilan pri gibanju niz nagib od 25°.

Kako bi algoritam za izbjegavanje bočnih zidova bio precizniji potrebno je ugraditi dva senzora udaljenosti koji bi zatvarali kut od 45° s prednjom i bočnom stranom robota, a bili bi smješteni na prednjem lijevom i prednjem desnom kutu robota. Također je potrebno ugraditi par stražnjih senzora udaljenosti kako bi se povećala robusnost autonomije robota.

Poželjno bi bilo da robot posjeduje vlastiti ekran na kojem bi imali uvid od trenutnim očitanjima pojedinih senzora udaljenosti, žiroskopa, te povratnih informacija o odlukama koje robot trenutno donosi, kao i vremena koje je proteklo od početka vožnje.



Slika 6.1 Prikaz prvotne razvojne verzije robota



Slika 6.2 Prikaz robota na osnovu kojeg je pisan diplomski rad

7 POPIS LITERATURE

- [1] Robo Cup Junior; „RoboCupJunior Rescue Maze – Rules 2020“ , s Interneta, https://junior.robotcup.org/wp-content/uploads/2020Rules/final_2020rules/Final-2020-Rescue-Maze1208.pdf, 16. rujna 2021.
- [2] Ivan Sičaja; „ BEŽIČNO UPRAVLJANI ROBOT TEMELJEN NA ARDUINO PLATFORMI “ , s Interneta, https://fesb-my.sharepoint.com/:b:/g/personal/isicaj00_fesb_hr/EaQfPzJTmnhAuJWQfHbvA0gBJ4PJ8pGrP_giW6Tvgv0DQw?e=aJY2d0, 16. rujna 2021.
- [3] PJRC; „ Teensy® 3.5 Development Bord “ , s Interneta, <https://www.pjrc.com/store/teensy35.html>, 16. rujna 2021.
- [4] Cult of Mac ; „Macintosh 512K“ , s Interneta, <https://www.cultofmac.com/500730/apple-history-fat-mac-quadruples-memory/>, 16. rujna 2021.
- [5] PJRC ; „Teensy Technical Specifications“ , s Interneta, <https://www.pjrc.com/teensy/techspecs.html>, 16. rujna 2021.
- [6] Etechnophiles; „Raspberry Pi 4 GPIO Pinout, Specs, Schematic (Detailed board layout)“ , s Interneta, <https://www.etechnophiles.com/raspberry-pi-4-gpio-pinout-specifications-and-schematic/>, 16. rujna 2021.
- [7] Sparkfun ; „Introduction to the Raspberry Pi GPIO and Physical Computing“ , s Interneta, <https://learn.sparkfun.com/tutorials/introduction-to-the-raspberry-pi-gpio-and-physical-computing/gpio-pins-overview>, 16. rujna 2021.
- [8] Helipal ; „storm 3,7v 180mah 15c mini li-po battery pack“ , s Interneta, <http://www.helipal.com/storm-3-7v-180mah-15c-mini-lipo-battery-pack.html>, 1. rujna 2021.
- [9] Hobbyking ; „Turnigy nano-tech 1800mah 3S 65~130C Lipo Pack“ , s Interneta, https://hobbyking.com/en_us/turnigy-nano-tech-1800mah-3s-65-130c-lipo-pack.html?__store=en_us, 16. rujna 2021.
- [10] Pololu; „156:1 Metal Gearmotor 20Dx44L mm 12V CB with Extended Motor Shaft“ , s Interneta, <https://www.pololu.com/product/3492>, 16. rujna 2021.
- [11] Microline; „Microline“ , s Interneta, <https://www.microline.hr/HTML.ashx?id=139>, 16. rujna 2021.
- [12] Robotshop ; „Sharp GP2Y0A21YK0F IR Range Sensor - 10cm to 80cm“ , s Interneta, <https://www.robotshop.com/uk/sharp-gp2y0a21yk0f-ir-range-sensor.html>, 16. rujna 2021.
- [13] How To Mechatronics; „How MEMS Accelerometer Gyroscope Magnetometer Work & Arduino Tutorial“ , s Interneta, https://www.youtube.com/watch?v=eqZ_gxR6eRjo, 16. rujna 2021.
- [14] Adafruit; „Adafruit BNO055 Absolute Orientation Sensor“ , s Interneta, <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/pinouts>, 16. rujna 2021.

- [15] Aliexpress; „TCS3200D TCS230 Color recognition sensor module“ , s Interneta, https://www.aliexpress.com/af/TCS3200D-TCS230-Color-recognition-sensor-module.html?tt=API&d=y&aff_platform=api-new-link-generate&origin=n&dp=7-1351&SearchText=TCS3200D+TCS230+Color+recognition+sensor+module&aff_fcid=48d83749bbcb47098be7749f1b984995-1631585120011-00812-_pRhTK2X&aff_fsk=_pRhTK2X&sk=_pRhTK2X&aff_trace_key=48d83749bbcb47098be7749f1b984995-1631585120011-00812-_pRhTK2X&terminal_id=d64195ae01ce420694d7ed3399e990e1, 16. rujna 2021.
- [16] Lastminuteengineers; „Interfacing TCS230/TCS3200 Color Sensor with Arduino“ , s Interneta, <https://lastminuteengineers.com/tcs230-tcs3200-color-sensor-arduino-tutorial/>, 16. rujna 2021.
- [17] Adafruit; „Adafruit AMG8833 8x8 Thermal Camera Sensor“ , s Interneta, <https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor>, 16. rujna 2021.
- [18] Generationrobots; „MCP3008 – 8-Channel 10-Bit ADC with SPI Interface“ , s Interneta, <https://www.generationrobots.com/en/402440-mcp3008-8-channel-10-bit-adc-with-spi-interface.html>, 16. rujna 2021.
- [19] Robotistan; „Raspberry Pi Camera Modul V2 - New Model“ , s Interneta, <https://www.robotistan.com/raspberry-pi-camera-modul-camera-modul-for-raspberry-pi>, 16. rujna 2021.
- [20] Raspberrypi; „Raspberry Pi Documentation“ , s Interneta, <https://www.raspberrypi.org/documentation/accessories/camera.html>, 16. rujna 2021.
- [21] Uctronics; „Arducam Multi Camera Adapter Doubleplexer Stereo Module V2 for Raspberry Pi Zero, Pi 3/3 b+, 4b“ , s Interneta, <https://www.uctronics.com/arducam-multi-camera-adapter-doubleplexer-stereo-module-v2-for-raspberry-pi-zero-pi-3-3-b-4b.html>, 16. rujna 2021.
- [22] Pcboard; „Mini Piezo Buzzer Mini PCB Mount“ , s Interneta, <https://www.pcboard.ca/minipiezo-buzzer>, 16. rujna 2021.
- [23] Cuidevices; „product spotlight: piezo and magnetic buzzers“ , s Interneta, <https://www.cuidevices.com/product-spotlight/piezo-and-magnetic-buzzers>, 16. rujna 2021.
- [24] Byjus; „light emitting diode“ , s Interneta, <https://byjus.com/physics/light-emitting-diode/>, 16. rujna 2021.
- [25] TLXTechnologies; „Solenoid Fundamentals: How Solenoids Work“ , s Interneta, <https://www.youtube.com/watch?v=Sq-CYfp9t4c>, 16. rujna 2021.
- [26] Alibaba; „Electric magnet TAU-0730B solenoid electromagnet pull&push stroke 10mm dc“ , s Interneta, <https://m.alibaba.com/product/1600113521483/Electric-magnet-TAU-0730B-solenoid-electromagnet.html>, 16. rujna 2021.

[27] Muthu; „All Tesseract OCR options“ , s Interneta, <https://muthu.co/all-tesseract-ocr-options/>, 16. rujna 2021.

[28] Dwrplastics; „1-2-1 Easy Mould 250g kit“ , s Interneta, <https://www.dwrplastics.com/1-2-1-easy-mould.html>, 16. rujna 2021.

[29] Pololu; „Pololu Universal Aluminum Mounting Hub for 4mm Shaft, #4-40 Holes (2-Pack)“ , s Interneta, <https://www.pololu.com/product/1081>, 16. rujna 2021.

[30] Mirko Pezo ; „Rescue Maze Robot - Programiranje“ , s Interneta, https://fesb-my.sharepoint.com/personal/mpezo01_fesb_hr/_layouts/15/onedrive.aspx?FolderCTID=0x01200096EE87E63B8B8B4F840E9CB12978B684&id=%2Fpersonal%2Fmpezo01_fesb_hr%2FDocuments%2FRescue%20Maze%20Robot%2FProgramiranje, 16. rujna 2021.

[31] Mirko Pezo ; „Rescue Maze Robot - Strojarsvo“ , s Interneta, https://fesb-my.sharepoint.com/personal/mpezo01_fesb_hr/_layouts/15/onedrive.aspx?FolderCTID=0x01200096EE87E63B8B8B4F840E9CB12978B684&id=%2Fpersonal%2Fmpezo01_fesb_hr%2FDocuments%2FRescue%20Maze%20Robot%2FStrojarsvo, 16. rujna 2021

SAŽETAK

U radu se opisuje postupak projektiranja robota prema kriterijima definiranim propisima Robo Cup Junior Rescue Maze 2020. Kriteriji za odabir svih komponenti su detaljno opisani, kao i principi funkcionalnosti pojedinih elektroničkih komponenti, programska logika, korišteni programski jezici i okruženja, te cijeli proces proizvodnje mehanike tijela robota. Rad opisuje eksperimentalnu provjeru funkcionalnosti robota i moguća poboljšanja u budućem radu.

KLJUČNE RIJEČI

Labirint, Robot, Arduino, Python, Obrada slike, Teensy 3.5, Raspberry Pi , Robo Cup Junior, Rescue Maze.

TITLE

THE PROJECTING AN AUTONOMOUS ROBOT OF THE RESCUE MAZE CATEGORY

SUMMARY

The thesis explains the procedure for designing robots according to the criteria defined by the Robo Cup Junior Rescue Maze 2020 regulations. The criteria for all component selection are described minutely as well as the principles of individual electronic components functionality, programming logic, used programming languages and frameworks, and the entire process of producing the robot body mechanics. The work describes experimental verification of the robot functionality and future work possible improvements.

KEY WORDS

Labirint, Robot, Arduino, Python, Image processing, Teensy 3.5, Raspberry Pi, Robo Cup Junior, Rescue Maze.

DODATAK A -PROGRAMSKI KOD

Budući da se programski kod sastoji od mnoštva zasebnih datoteka, te je vrlo opširan, dostupan je na referenci [30] umjesto u tekstualnom obliku u diplomskom radu.